

2

ADA136779

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
JAN 13 1984

THESIS

DTIC FILE COPY

A MANAGEMENT INFORMATION SYSTEM
FOR THE ANALYSIS OF THE
ARMED SERVICES APTITUDE BATTERY

by

Robert Joseph Forman

September 1983

Thesis Advisor:

D.R. Dolk

Approved for public release; distribution unlimited.

84 01 13 113

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD A13 6779	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) A Management Information System for the Analysis of the Armed Services Aptitude Battery		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1983	
		6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) Robert Joseph Forman		8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1983	
		13. NUMBER OF PAGES 141	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		Accession For NTIS GRA&I <input checked="" type="checkbox"/>	
		DTIC TAB <input type="checkbox"/> Unannounced <input type="checkbox"/> Justification	
18. SUPPLEMENTARY NOTES		By _____	
		Distribution _____	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Armed Services Vocational Aptitude Battery (ASVAB) Query Language Processor, Series 1100 (QLP 1100) Data Base Management Systems		Dist _____	
		A-1	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → This thesis develops a management information system called ASVABMIS for use by Army agencies conducting analysis using data from the Armed Services Aptitude Battery. The thesis develops a FORTRAN program that inputs demographic data and raw scores, and computes subtest standard scores, and the Army Classification Battery Composites. The output is directed to files designed to be			

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE
5/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

easily used with the Statistical Package for the Social Sciences for statistical analysis. In addition, a data base management system is integrated into the information system for data organization and management.

Approved for public release; distribution unlimited.

A Management Information System
for the Analysis of the
Armed Services Aptitude Battery

by

Robert Joseph Forman
Captain, United States Army
B.S., United States Military Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL

September 1983

Author:

Robert J. Forman

Approved by:

Daniel R. Dolk

Thesis Advisor

Stephen J. Paul

Second Reader

Richard L. Elster

Chairman, Department of Administrative Science

K.T. Marshall

Dean of Information and Policy Sciences

ABSTRACT

This thesis develops a management information system called ASVABMIS for use by Army agencies conducting analysis using data from the Armed Services Aptitude Battery. The thesis develops a FORTRAN program that inputs demographic data and raw scores, and computes subtest standard scores, and the Army Classification Battery Composites. The output is directed to files designed to be easily used with the Statistical Package for The Social Sciences for statistical analysis. In addition, a data base management system is integrated into the information system for data organization and management.

TABLE OF CONTENTS

I.	INTRODUCTION -----	9
	A. PURPOSE -----	9
	B. PROBLEM -----	9
	C. FEASIBILITY -----	10
	D. SYSTEM DESIGN -----	10
II.	SYSTEMS MANUAL -----	13
	A. PURPOSE -----	13
	B. INTRODUCTION -----	13
	C. THE ASVAB 5 PROCEDURE -----	18
	D. THE ASVAB 6 AND ASVAB 7 PROCEDURE -----	23
	E. THE ASVAB 8, ASVAB 9, AND ASVAB 10 PROCEDURE -----	28
III.	USER'S MANUAL -----	33
	A. PURPOSE -----	33
	B. INTRODUCTION -----	33
	C. THE INPUT FILES -----	33
	D. THE ALGORITHM -----	35
	E. THE OUTPUT FILES -----	47
IV.	SPSS AND QLP 1100 -----	49
	A. PURPOSE -----	49
	B. INTRODUCTION -----	49
	C. SPSS -----	50
	D. QLP 1100 -----	54

1. File Definition Processor Source Input -----	56
2. Application Definition Processor Source Input -----	64
3. FDP and ADP Call Commands -----	66
4. Query Language Processor, 1100 Series ---	71
5. Summary Example -----	77
V. CONCLUSIONS -----	81
APPENDIX A: INPUT DATA FILE -----	83
APPENDIX B: ASVAB 5/6/7 SSS CONVERSION TABLE -----	84
APPENDIX C: ASVAB 8/9/10 SSS CONVERSION TABLE -----	85
APPENDIX D: ASVAB 5 COMPOSITE CONVERSION TABLE -----	87
APPENDIX E: ASVAB 6/7 COMPOSITE CONVERSION TABLE -----	92
APPENDIX F: ASVAB 8/9/10 COMPOSITE CONVERSION TABLE --	94
APPENDIX G: ASVAB PROGRAM OUTPUT FILES -----	99
APPENDIX H: SPSS REGRESSION PROGRAM LISTING -----	105
APPENDIX I: SPSS REGRESSION MODEL PROGRAM LISTING -----	106
APPENDIX J: SPSS REGRESSION PROGRAM OUTPUT -----	107
APPENDIX K: SPSS REGRESSION MODEL PROGRAM OUTPUT -----	109
APPENDIX L: ASVAB PROGRAM LISTING -----	110
BIBLIOGRAPHY -----	140
INITIAL DISTRIBUTION LIST -----	141

TABLE OF ABBREVIATIONS

ACB	Army Classification Battery
AD	Attention To Detail
ADP	Application Definition Processor
AI	Automotive Information
AR	Arithmetic Reasoning
AS	Automotive/Shop
ASVAB	Armed Services Vocational Aptitude Battery
ASVABMIS	Armed Services Vocational Aptitude Battery Management Information System
CA	Attentiveness
CC	Combat
CE	Electronics
CL	Clerical
CM	Maintenance
CO	Combat
CS	Coding Speed
DOES	Department of Evaluation and Standardization
EI	Electronics Information
EL	Electronics Repair
FA	Field Artillery
FDP	File Definition Processor
GI	General Information
GM	General Maintenance
GS	General Science

GT	General Technical
MC	Mechanical Comprehension
MK	Math Knowledge
MM	Motor Maintenance
NO	Numerical Operations
OF	Operators and Food Handlers
PC	Paragraph Comprehension
PCIOS	Processor Common Input/Output System'
QLP	Query Language Processor
SC	Surveillance and Communications
SI	Shop Information
SP	Space Perception
SPSS	Statistical Package for the Social Sciences
SQT	Skill Qualification Test
SSS	Subtest Standard Score
ST	Skilled Technician
VE	Verbal
WK	Word Knowledge

I. INTRODUCTION

A. PURPOSE

The purpose of this chapter is to discuss a problem faced by the Department of Evaluation and Standardization (DOES), Fort Benning, Georgia. In addition, this chapter will determine the feasibility of providing DOES with a computer-based management information system. Finally, this chapter will conceptually develop a system design that will meet DOES' needs.

B. PROBLEM

DOES is responsible for evaluating and standardizing training programs developed by the Infantry School, Fort Benning, Georgia. DOES uses the raw scores that are a part of the Armed Services Aptitude Battery in conducting analysis of many of the training programs. Currently, DOES uses a manual system that requires a minimum of 21 entries into a conversion table, as well as the computation of 10 equations. This system becomes prohibitively time consuming with large sample sizes. For example, DOES is responsible for developing a selection criteria for the Infantry School's mechanized infantryman training program. The school trains approximately 100 soldiers during each training cycle which lasts only a few weeks. In order to process the data, and to develop the

selection criteria using the manual system, DOES would have to allocate much of its resources in terms of manpower and time to this project alone. Since DOES is unable to do this, the selection criteria has not been developed, and a large backlog of data exists which must be processed. Thus a speedier system is needed that can process the backlog as well as any new data using, at most, one analyst rather than the entire analytical manpower pool. ASVABMIS is the speedier system that will allow DOES to accomplish all of its requirements.

C. FEASIBILITY

The feasibility of a computer-based management information system is apparent in surveying the hardware and software available to DOES. DOES has access to a UNIVAC 1100 main-frame computer with a FORTRAN compiler. In addition, the Statistical Package for the Social Sciences is available for statistical analysis, while the SPERRY-UNIVAC QLP 1100 system is available for data management. ASVAB raw data is provided in the form of prepunched computer cards. The only component that is lacking is a software program to provide DOES with a complete system for data generation, analysis, and management.

D. SYSTEM DESIGN

In designing the software, DOES has some specific criteria that must be met. First, DOES currently has only batch

processing available. Thus the program cannot be interactive. Secondly, DOES requires that the program be written in FORTRAN because of the availability of the FORTRAN compiler. Since DOES will use SPSS in batch mode, any data files generated by the program must be compatible with the SPSS format. Finally, DOES has varying needs in regard to report formats and other data management. Thus, it is difficult, if not impossible to anticipate every need in order to code every report format in the program itself. This leads to the necessity of introducing data management and organization through a data base management system. Therefore, the file definitions and query language of the QLP 1100 system must be developed in conjunction with DOES' needs.

In summary, the information system must be comprised of a FORTRAN program that inputs data, computes the necessary ASVAB scores, and generates data to be stored in a file. The files must be structured so that they provide immediate, useful information to DOES. In addition, the files must be in a form that allows integration into both SPSS for statistical analysis and QLP 1100 for data management. This design will provide DOES with both the flexibility to generate the required information in the appropriate format and provide linkages to powerful analytic packages.

Chapter I, "Introduction," has discussed one of the problems faced by DOES. In addition, it has examined the feasibility of a computer-based management information system.

Finally, the chapter has identified the user's performance specifications for the information system. Chapter II, "Systems Manual," examines the manual system in detail. It provides several examples and is designed to familiarize readers with the current system. Chapter III, "User's Manual," explains the computer program using nontechnical language. It is designed to provide the reader with a conceptual development of the computer program. Chapter IV, "The SPSS and QLP 1100 System," explains how the FORTRAN program is integrated with the SPSS program and the QLP 1100 system. Chapter V, "Conclusions," explains how the information system and its use may be expanded.

II. SYSTEMS MANUAL

A. PURPOSE

The purpose of this chapter is to provide an understanding of how the current manual system works. This is important because the procedures used in the computerized system are virtually the same.

B. INTRODUCTION

The Armed Services Vocational Aptitude Battery has six test versions that can be grouped into three different test formats. The first group is comprised of ASVAB 5. The second group is comprised of ASVAB 6 and ASVAB 7. The last group is comprised of ASVAB 8, ASVAB 9 and ASVAB 10.

The manual process used by DOES is shown in Figure 1. The input data are stored on pre-punched computer cards (see Appendix A). This data stream is decoded using the key shown in Table 1.

Once the data are decoded, they can be checked for input errors. This is important because DOES does not enter the data onto the cards, and must confirm that they are correct. Next the raw scores are converted to standard scores. This is done so that scores can be compared between test versions. Composite scores are then computed, and subsequently normalized so that they can also be compared between test

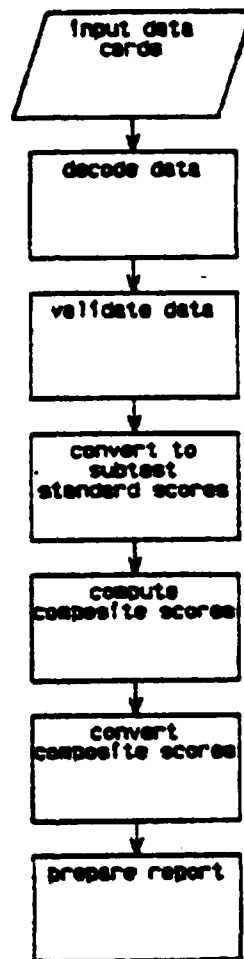


Figure 1. ASVAB Manual Process

TABLE 1
INPUT DATA KEY

<u>COLUMNS</u>	<u>DESCRIPTION</u>
1-9	Social Security Account Number
10-15	Date of Birth (YY/MM/DD)
16	Sex: 1-Male, 2-Female
17	Race: 1-White, 2-Black, 3-Other
18-23	Date of Entry (YY/MM/DD)
24-25	Highest Year of Education:
	01 1-7 Years
	02 8 Years
	03 1 Year of High School
	04 2 Years of High School
	05 3-4 Years of High School
	06 High School Diploma
	07 1 Year of College
	08 2 Years of College
	09 3-4 Years of College
	10 College Graduate
	11 Master's Degree
	12 Doctor's Degree
	13 High School GED
26-27	Entry Pay Grade:
	01 E1
	02 E2
	03 E3

TABLE 1 (CONT.)

04 E4

05 E5

06 E6

07 E7

08 E8

09 E9

28-32 Military Occupation Specialty

33-34 Test Form:

35 ASVAB 5.

36 ASVAB 6.

37 ASVAB 7.

38 ASVAB 8.

39 ASVAB 9.

40 ASVAB 10.

35-36 AFQT Percentile

37-68 Subtest Raw Scores:

	ASVAB 5	ASVAB 6/7	ASVAB 8/9/10
37-38	GI	GI	GS
39-40	NO	NO	AR
41-42	AD	AD	WK
43-44	WK	WK	PC
45-46	AR	AR	NO
47-48	SP	SP	CS
49-50	MK	MK	AS
51-52	EI	EI	MK
53-54	MC	MC	MC
55-56	GS	GS	EI

TABLE 1 (CONT.)

57-58	SI	SI	VE
59-60	AI	AI	BLANK
61-62	BLANK	MC	BLANK
63-64	BLANK	CA	BLANK
65-66	BLANK	CE	BLANK
67-68	BLANK	CC	BLANK
71-73	SEPARATION PROGRAM DESIGNATOR.		
74-79	DATE OF SEPARATION.		

versions. Finally, the data are written down in a format that is conducive to further analysis. This process is done for each data card. A typical session could involve 100 to 300 data cards depending on the size of the training group. Also the frequency of this process depends on the number of training cycles being run by the Infantry School.

C. THE ASVAB 5 PROCEDURE

In order to process an ASVAB 5 record, the data record is first decoded using Table 1. The first data record in Appendix A is provided as a typical ASVAB 5 data card. The number 35 located in columns 33 and 34 indicates that ASVAB 5 is in effect. The scores located in columns 37 through 68 can be identified and assigned to specific tests. This assignment is shown in Table 2.

Since ASVAB 5 is in effect, these raw scores must be converted into subtest standard scores. The ASVAB 5/6/7 subtest standard score conversion table is used to convert the scores (see Appendix B). The table is entered by finding the raw score in the left hand column, then moving right to find the test name and the appropriate converted score. For example, to convert GI enter the raw score column at 15, move to the right and find the column for GI and read off the standard score of 66. Proceeding in this manner, all subtest raw scores are converted as shown in Table 3.

The Army Classification Battery (ACB) uses subtest standard scores to compute the equations shown in Table 4.

TABLE 2

ASVAB 5 RAW SCORE ASSIGNMENT

<u>SUBTEST (Variable Name)</u>	<u>SCORE (Data Point)</u>
GI	15
NO	20
AD	03
WK	12
AR	06
SP	17
MK	20
EI	10
MC	09
GS	05
SI	13
AI	12

TABLE 3
SUBTEST SCORE CONVERSIONS

<u>RAW SCORE</u>	<u>SUBTEST</u>	<u>STANDARD SCORE</u>
15	GI	66
20	NO	39
03	AD	21
12	WK	39
06	AR	36
17	SP	59
20	MK	67
10	EI	34
09	MC	46
05	GS	36
13	SI	48
12	AI	50

TABLE 4
ASVAB 5 ACB EQUATIONS

<u>EQUATION</u>	<u>SCORE</u>
CO = AR + SI + SP + AD	164
FA = AR + GI + MK + EI	203
MM = MK + SI + EI + AI	199
GM = AR + GS + MC + AI	168
CL = AR + WK + AD	96
GT = AR + WK	75
EL = AR + EI + NC + SI	164
SC = AR + WK + MC + SP	180
ST = AR + MK + GS	139
OF = GI + AI	116

The final step is to convert the ACB scores to standardized scores using the ASVAB 5 composite conversion table (see Appendix D). The table is used exactly as the ASVAB 5 subtest standard score conversion table to give the final composite scores shown in Table 5.

TABLE 5
ASVAB 5 ACB COMPOSITE SCORES

<u>ACB SCORE</u>	<u>CLASSIFICATION</u>	<u>COMPOSITE SCORE</u>
164	CO	71
203	FA	102
199	MM	99
168	GM	79
96	CL	49
75	GT	71
164	EL	76
180	SC	89
139	ST	94
116	OF	116

With the calculations completed, the soldier file can be constructed using the results of the calculations and the initial data record. The soldier file for the example data record is shown below.

SSAN: 272598843

DATE OF BIRTH(YY/MM/DD): 64/4/20

SEX: MALE

RACE: BLACK

DATE OF ENTRY(YY/MM/DD): 82/2/16

HIGHEST YEAR OF EDUCATION: 1 YEAR OF HIGH SCHOOL

ENTRY PAY GRADE: E1

MOS: 11B10

TEST FORM: ASVAB 5

AFQT PERCENTILE: 35

SEPARATION PROGRAM DESIGNATOR: AKL

DATE OF SEPARATION(YY/MM/DD) 83/2/27

SUBTEST RAW SCORES

GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI
15	20	03	12	06	17	20	10	09	05	13	12

SUBTEST STANDARD SCORES

GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI
66	39	21	39	36	59	67	34	46	36	48	50

ARMY CLASSIFICATION BATTERY

CO	FA	MM	GM	CL	GT	EL	SC	ST	OF
71	102	99	79	49	71	76	89	94	116

D. THE ASVAB 6 AND ASVAB 7 PROCEDURE

The procedures for ASVAB 6 and ASVAB 7 are the same.

However, these procedures differ significantly from the ASVAB

5 procedures. The first difference is that, in addition to the twelve raw scores used in ASVAB 5, four additional scores, CM, CA, CE, and CC are also used. The second difference is that the subtest raw scores are not converted to standard scores for the purpose of computing composites as they were in the ASVAB 5 procedure. They are converted for comparison with scores from other test versions. Finally, the set of equations used to compute the ACB composites are different. A typical input data record for ASVAB 6 or ASVAB 7 is provided as the second data record in Appendix A.

Since there is no need to convert the subtest raw scores, the procedure begins with the computation of the ACB composites. First, the data in columns 33 and 34 are checked to confirm that ASVAB 6 or ASVAB 7 is in effect. Using the sample input data record columns 33 and 34 have the value 36 as a data point confirming ASVAB 6. The data in columns 37 through 68 are extracted and assigned to the appropriate subtests as shown in Table 6.

Since raw scores are used, the ACB composites are immediately computed as shown in Table 7.

The final step is to convert the ACB scores to standardized scores using the ASVAB 6/7 composite conversion table (see Appendix E) as shown in Table 8.

With the conversion of the ACB scores to standard composite scores a soldier file can be constructed. However, one additional step is done. As shown above, the raw subtest

TABLE 6
ASVAB 6/7 RAW SCORE ASSIGNMENT

<u>SUBTEST (Variable Name)</u>	<u>SCORE (Data Point)</u>
GI	13
NO	40
AD	25
WK	22
AR	15
SP	16
MK	12
EI	23
MC	10
GS	09
SI	17
AI	14
CM	08
CA	12
CE	13
CC	22

TABLE 7
ASVAB 6/7 ACB EQUATIONS

<u>EQUATION</u>	<u>SCORE</u>
CO = AR + SI + SP + AD + CC	95
FA = AR + GI + MK + EI + CA	75
MM = MK + SI + EI + AI + CM	74
GM = AR + GS + MC + AI	48
CL = AR + WK + AD + CA	74
GT = AR + WK	37
EL = AR + EI + MC + ST + CE	78
SC = AR + WK + MC + SP	63
ST = AR + MK + GS	36
OF = GI + AI + CA	39

TABLE 8
ASVAB 6/7 ACB COMPOSITE SCORES

<u>ACB SCORE</u>	<u>CLASSIFICATION</u>	<u>STANDARD SCORE</u>
95	CO	125
75	FA	115
74	MM	106
48	GM	103
74	CL	122
37	GT	109
78	EL	114
63	SC	108
36	ST	104
39	OF	115

scores are used in computing the ACB scores. These raw scores cannot be compared to the standardized scores in other test versions. Because of this, the raw scores are converted using the ASVAB 5/6/7 standard score conversion table (see Appendix B).

The input data record and the ACB computations are used to create the soldier file.

SSAN: 016532241

DATE OF BIRTH(YY/MM/DD): 63/11/25

SEX: MALE

RACE: WHITE

DATE OF ENTRY(YY/MM/DD): 81/9/20

HIGHEST YEAR OF EDUCATION: HIGH SCHOOL DIPLOMA

ENTRY PAY GRADE: E1

MOS: 11B10

TEST FORM: ASVAB 6

AFQT PERCENTILE: 87

SEPARATION PROGRAM DESIGNATOR: AAA

DATE OF SEPARATION(YY/MM/DD): 82/11/9

SUBTEST RAW SCORES

GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI	CM	CA	CE	CC
13	40	25	22	15	16	12	23	10	09	17	14	08	12	13	22

SUBTEST STANDARD SCORES

GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI	CM	CA	CE	CC
60	59	77	53	55	57	51	56	48	45	58	55	-	-	-	-

ARMY CLASSIFICATION BATTERY

CO	FA	MM	GM	CL	GT	EL	SC	ST	OF
125	115	106	103	122	109	114	108	104	115

E. ASVAB 8, ASVAB 9, AND ASVAB 10 PROCEDURE

The procedure for ASVAB 8, ASVAB 9, and ASVAB 10 is generally the same as ASVAB 5. The subtest raw scores are converted to subtest standard scores. The converted scores are used to compute the ACB composite scores. Finally, the ACB scores are standardized. The difference between ASVAB 8, ASVAB 9, and ASVAB 10, and all other ASVAB versions is that a different set of tests are used to generate the raw scores. Because of this, different conversion tables are used, as well as a different set of equations for the computation of the ACB scores. A typical input data record for ASVAB 8, ASVAB 9, or ASVAB 10 is provided as the third data card in Appendix A.

First, columns 33 and 34 are checked to insure that either ASVAB 8, ASVAB 9, or ASVAB 10 is in effect. The data point 40 confirms that ASVAB 10 is in effect. The raw scores in columns 37 through 58 are extracted and assigned to their appropriate subtest variable. The raw scores are then standardized using the ASVAB 8/9/10 subtest standard score conversion table (see Appendix C) as shown in Table 9.

The subtest standard scores are then used to compute the ACB scores as shown in Table 10.

TABLE 9

ASVAB 8/9/10 SUBTEST SCORE CONVERSIONS

<u>RAW SCORE (Data Point)</u>	<u>SUBTEST (Variable Name)</u>	<u>STANDARD SCORE</u>
24	GS	65
22	AR	56
30	WK	56
14	PC	60
45	NO	59
79	CS	72
25	AS	65
20	MK	63
24	MC	65
19	EI	65
48	VE	61

TABLE 10
ASVAB 8/9/10 ACB EQUATIONS

<u>EQUATION</u>	<u>SCORE</u>
CO = AR + AS + MC + CS	258
FA = AR + MK + MC + CS	256
MM = NO + EI + MC + AS	254
GM = MK + EI + GS + AS	258
CL = NO + CS + VE	192
GT = VE + AR	117
EL = AR + EI + MK + GS	249
SC = NO + CS + VE + AS	257
ST = VE + MK + MC + GS	254
OF = NO + VE + MC + AS	250

Finally, once the ACB scores are computed, the scores are standardized using the ASVAB 8/9/10 composite conversion table (see Appendix F) as shown in Table 11.

TABLE 11
ASVAB 8/9/10 ACB COMPOSITE SCORES

<u>ACB SCORE</u>	<u>CLASSIFICATION</u>	<u>STANDARD SCORE</u>
258	CO	138
256	FA	128
254	MM	138
258	GM	132
192	CL	133
117	GT	117
249	EL	124
257	SC	144
254	ST	128
250	OF	135

The input data record and the ACB computations are used to generate the following soldier file.

SSAN: 213865527

DATE OF BIRTH(Y Y/MM/DD): 630521

SEX: FEMALE

RACE: OTHER

DATE OF ENTRY(Y Y/MM/DD): 821007

HIGHEST YEAR OF EDUCATION: 1 YEAR OF COLLEGE

ENTRY PAY GRADE: E2

MOS: 54E10

TEST FORM: ASVB 10

AFQT PERCENTILE: 92

SUBTEST RAW SCORES

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
24	22	30	14	45	79	25	20	24	19	48

SUBTEST STANDARD SCORES

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
65	56	56	60	59	72	65	63	65	65	61

ARMY CLASSIFICATION BATTERY

CO	FA	MM	GM	CL	GT	EL	SC	ST	OF
138	128	138	132	133	117	124	144	128	135

This process continues until all the input data records are processed into soldier records.

III. USER'S MANUAL

A. PURPOSE

The purpose of this chapter is to provide the user of the Armed Services Vocational Aptitude Battery Management Information System (ASVABMIS) with a manual that explains the functioning of the ASVAB program. The language is non-technical, yet specific enough to allow a thorough understanding or how ASVAB works.

B. INTRODUCTION

The relationship between ASVAB and its input file and output files is shown in Figure 2. Figure 2 shows that separate input data cards are combined into one input data file. The input data file is processed by ASVAB, and four output files called the soldier file, the raw score summary file, the standard score summary file, and the composite score summary file result.

C. THE INPUT FILES

In order to create the input data file the user must simply read into file 04 input data, the stack of input computer cards. No special preparation is necessary, and this must be done for each different set of input cards. The data cards containing the conversion tables must be read into file and permanently stored when ASVAB is installed.

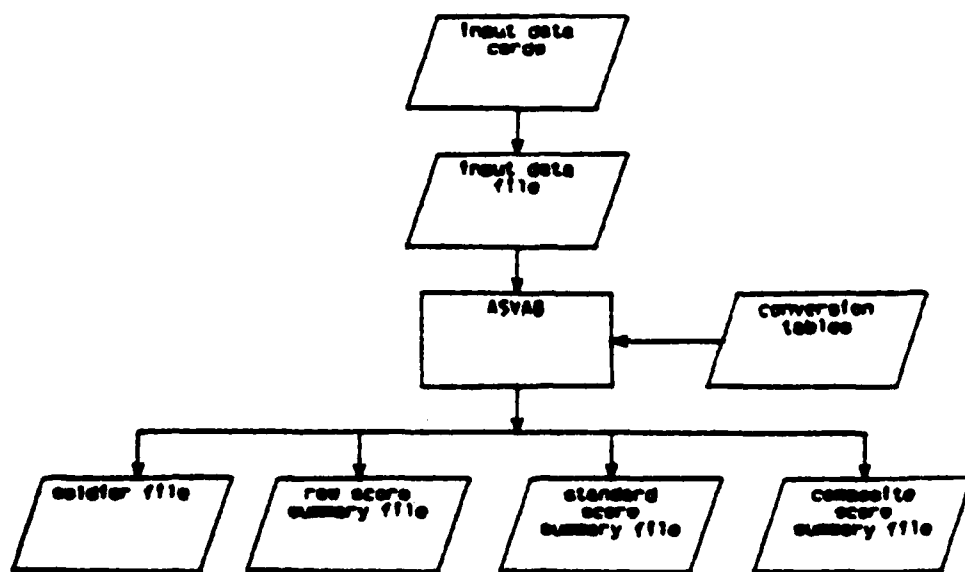


Figure 2. Relationship Between ASVAB Program and Input/Output Files

The stack of cards containing the conversion tables for ASVAB 5, ASVAB 6/7, and ASVAB 8/9/10 are read into File 01, File 02, and File 03 respectively.

D. THE ALGORITHM

The development of the system began with the determination of what ASVAB had to accomplish. This was done by studying the manual system. The initial algorithm was developed as shown below.

ALGORITHM ASVAB

READ IN CONVERSION TABLES

ALGORITHM INPUT SOLDIER DATA

READ IN INPUT DATA

END ALGORITHM INPUT SOLDIER DATA

ALGORITHM VALIDATE SOLDIER DATA

CHECK FOR DATA INTEGRITY

END ALGORITHM VALIDATE SOLDIER DATA

ALGORITHM OUTPUT SOLDIER DATA

PRINT THE INPUT DATA AS PART OF THE SOLDIER FILE

END ALGORITHM OUTPUT SOLDIER DATA

ALGORITHM SUBTEST STANDARD SCORES

CONVERT RAW SCORES TO SUBTEST STANDARD SCORES

END ALGORITHM SUBTEST STANDARD SCORES

ALGORITHM APTITUDE AREA COMPOSITES

COMPUTE ACB COMPOSITES

END ALGORITHM APTITUDE AREA COMPOSITES

ALGORITHM ACB

CONVERT ACB COMPOSITES

END ALGORITHM ACB

ALGORITHM OUTPUT SCORES

PRINT SUBTEST STANDARD SCORES AND ACB SCORES

END ALGORITHM OUTPUT SCORES

END ALGORITHM ASVAB

The relationship between the algorithms is shown in Figure 3.

1. Algorithm ASVAB

This algorithm defines the variables that are used in ASVAB. It also reads in the conversion tables, and terminates processing when the last data card has been read.

2. Algorithm Input Soldier Data

This algorithm reads in the data from File 04 input data. The process is shown in Figure 4. Figure 3 shows that once file 04, Input Data, is read in, a decision logic is used to assign the raw scores to their proper variable names.

3. Algorithm Validate Soldier Data

This algorithm provides a rudimentary input data validation by making range checks on the data according to Table 12. If the value of a variable is outside the range shown in Table 12, an error message is printed. In general, the error statements identify the record that has

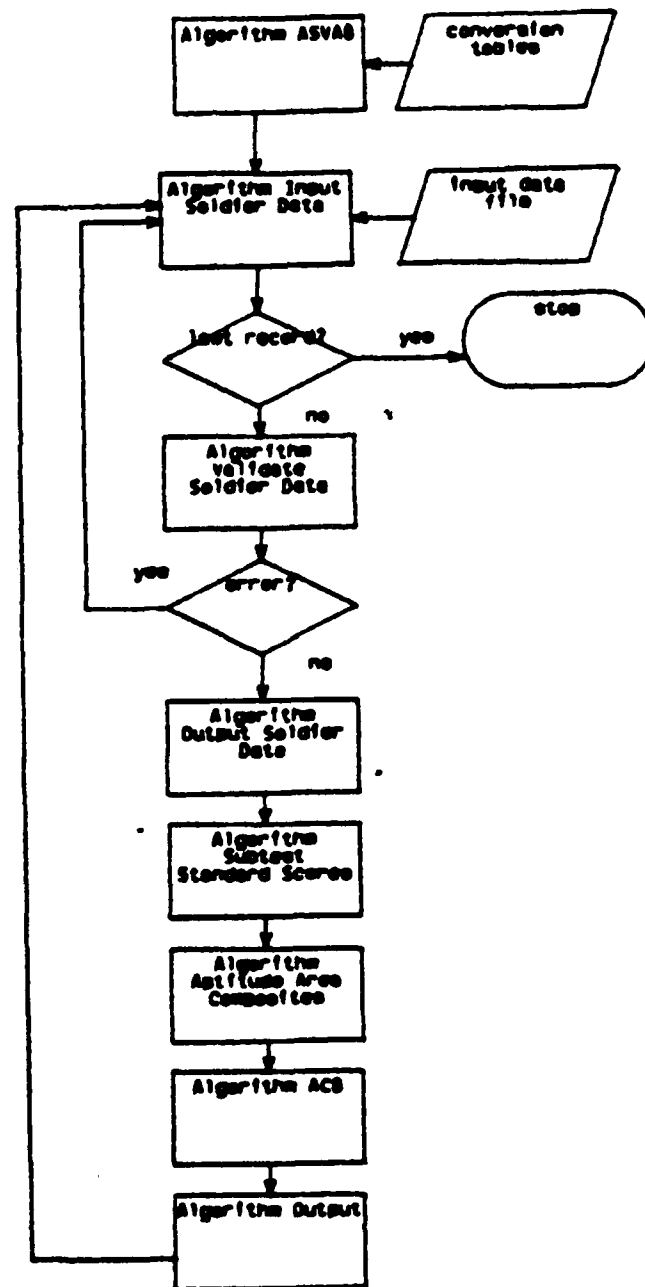


Figure 3. ASVAB Program Flow Chart

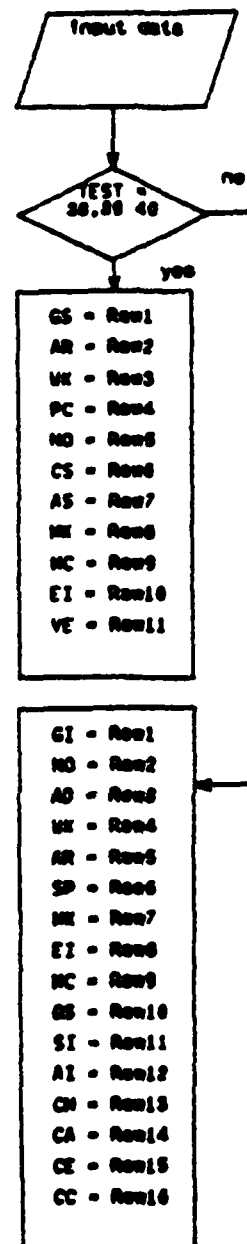


Figure 4. Algorithm Input Soldier Data Flow Chart

TABLE 12
DATA INTEGRITY TABLE

<u>ITEM</u>	<u>MIN. VALUE</u>	<u>MAX. VALUE</u>
MONTH (MM)	1	12
DAY (DD)	1	31
SEX	1	2
RACE	1	3
ED	1	13
TEST	35	40
ASVAB 5/6/7		
GI	0	15
NO	0	30
AD	0	30
WK	0	20
AR	0	20
SP	0	20
MK	0	20
EI	0	30
MC	0	20
GS	0	20
SI	0	20
AI	0	20
CM	0	20
CA	0	20
CE	0	20
CC	0	27

TABLE 12 (CONT.)

<u>ITEM</u>	<u>MIN. VALUE</u>	<u>MAX. VALUE</u>
ASVAB 8/9/10		
GS	0	25
AR	0	30
WK	0	35
PC	0	15
NO	0	50
CS	0	84
AS	0	25
MK	0	25
MC	0	25
EI	0	25
VE	0	50

the error by record number, the social security number, the incorrect variable and its column position in the record. Processing of that record is terminated and processing of a new record is begun.

4. Algorithm Output Soldier Data

This algorithm simply prints the input data. This output is provided for two reasons. First, it provides some of the information required by the user in an understandable form. Second, since the data is echoed, error messages can be confirmed. The algorithm provides the information as shown below.

SSAN	:	014560821
DATE OF BIRTH(Y Y/M M/D D)	:	64/4/20
SEX	:	1-MALE
RACE	:	2-BLACK
DATE OF ENTRY(Y Y/M M/D D)	:	82/2/16
HIGHEST YEAR OF EDUCATION	:	1 YEAR OF HIGH SCHOOL--03
ENTRY PAY GRADE: 01-E1	:	01-E1
MOS	:	11B10
TEST FORM	:	40-ASVAB 10
AFQT PERCENTILE	:	89
SEPARATION PROGRAM DESIGNATOR:		JHK
DATE OF SEPARATION(Y Y/M M/D D)	:	83/5/25

SUBTEST RAW SCORES

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
20	15	10	05	35	60	10	20	13	06	35

5. Algorithm Subtest Standard Scores

This algorithm converts the subtest raw scores into subtest standard scores. Instead of using the conversion tables to convert the raw scores, the standard scores are computed directly using the equation

$$SSS = \left[10 \times \frac{(\text{RAW SCORE} - \bar{X})}{\sigma} \right] + 50$$

The raw scores are read from input data. The expected value and the standard deviation for each subtest and ASVAB version have been calculated previously and are provided in the ASVAB program. Since the conversions must be done repeatedly, a function is used to calculate the conversions. This algorithm merely invokes the function for each subtest.

For example, if ASVAB 5 is in effect, to convert a raw score of 10 for GI the function is invoked by `GISSS = CONVT(GI,XBAR,SIGMA)`. An actual computation is given below.

$$GISSS = \text{CONVT}(10, 9.656, 3.273)$$

$$GISSS = \left[10 \times \frac{(10 - 9.656)}{3.273} \right] + 50$$

$$GISSS = 51.05$$

The function rounds down to the nearest integer if the decimal value is less than 0.50, and rounds up if the decimal value

is greater than or equal to 0.50. Since 0.05 is less than 0.50, GISSS = 51. This is the same value obtained using the conversion tables. This method is more efficient and requires less storage than the conversion table method. A decision table is used to determine which ASVAB version is in effect so that the correct raw scores, expected values, and standard deviations are used.

6. Algorithm Aptitude Area Composites

This algorithm uses the appropriately converted subtest scores from the previous algorithm to compute the Army Aptitude Composites. A decision table based on the value of TEST is used to determine which set of equations will be used. The equations for ASVAB 5, ASVAB 6/7, and ASVAB 8/9/10 are given in Tables 4, 7, and 10 respectively. This process is shown in Figure 5.

7. Algorithm ACB

This algorithm converts the scores computed in the previous algorithm into standard scores that comprise the Army Classification Battery. The process used is the table look-up process used in the manual system.

The computer stores in memory the conversion tables for ASVAB 5, ASVAB 6/7, and ASVAB 8/9/10. The three tables are stored as $m \times n$ matrices called A5, A67, and A8910, respectively. The variable m corresponds to the maximum number of rows in the matrix. Therefore, the maximum value of m must equal the maximum raw score possible. The variable n

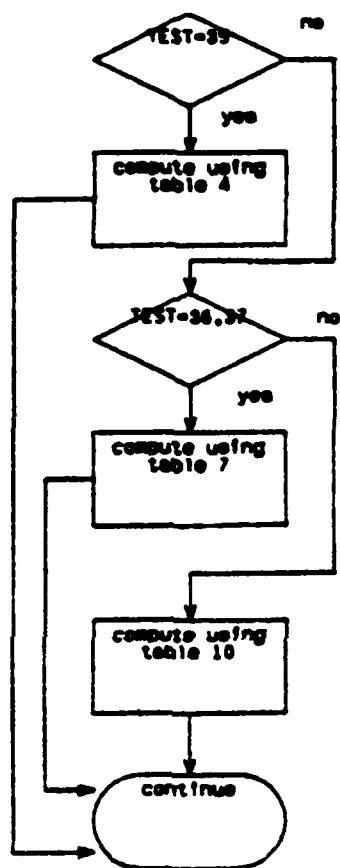


Figure 5. Algorithm Aptitude Area Composites Flow Chart

corresponds to the maximum number of columns in the matrix. Therefore, the maximum value of n must be 10 since there are 10 different scores to be converted. Table 13 shows the column assignments.

TABLE 13
MATRIX COLUMN ASSIGNMENTS

<u>n</u>	<u>SCORE</u>
1	GT
2	GM
3	EL
4	CL
5	MM
6	SC
7	CO
8	FA
9	OF
10	ST

For example, if the value of TEST is 38, 39, or 40 then ASVAB 8, 9, or 10 respectively is in effect. The algorithm makes the following assignments:

GT = A8910(GT,1)

GM = A8910(GM,2)

EL = A8910(EL,3)

CL = A8910(CL,4)

MM = A8910(MM,5)
 SC = A8910(SC,6)
 CO = A8910(C),7)
 FA = A8910(FA,8)
 OF = A8910(OF,9)
 ST = A8910(ST,10)

The same procedure is used for ASVAB 5, and ASVAB 6/7, except that matrices A5 and A67 are used respectively.

8. Algorithm Output Scores

The purpose of this algorithm is to print the subtest standard scores, and the converted ACB composite scores. The algorithm output soldier data printed the following information:

SSAN	:	014560821
DATE OF BIRTH(YY/MM/DD)	:	64/4/20
SEX	:	1-MALE
RACE	:	2-BLACK
DATE OF ENTRY(YY/MM/DD)	:	82/2/16
HIGHEST YEAR OF EDUCATION	:	1 YEAR OF HIGH SCHOOL-03
ENTRY PAY GRADE	:	01-E1
MOS	:	11B10
TEST FORM - ASVAB 10	:	40-ASVAB 10
AFQT PERCENTILE	:	89
SEPARATION PROGRAM DESIGNATOR:		JHK
DATE OF SEPARATION(YY/MM/DD)	:	83/5/25

SUBTEST RAW SCORES

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
20	15	10	05	35	60	10	20	13	06	35

This algorithm adds the following lines giving the final soldier file.

SUBTEST STANDARD SCORES

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
57	46	30	34	49	60	39	63	45	35	49

ARMY CLASSIFICATION BATTERY

CO	FA	MM	GM	CL	GT	EL	SC	ST	OF
93	108	78	95	102	93	101	94	107	86

In addition, the subtest raw scores, the subtest standard scores, and the ACB composite scores are each printed in a separate summary file.

E. THE OUTPUT FILES

ASVAB generates four output files. The soldier file is stored in File 9. The soldier file is the most important file and should be printed immediately by the user (see Appendix G). Invalid input data cards will be identified by error messages in the file. This allows the user to make corrections to input data.

In addition three summary files are generated (see Appendix G). A summary of the standard scores, the raw

scores, and the composite scores are stored in Files 10, 11 and 12 respectively. The summary files do not contain any information that is not already contained in the soldier file. The summary files are not generated to provide information in paper copy form, but are generated in tabular form to provide input into SPSS and QLP 1100.

IV. SPSS AND QLP 1100

A. PURPOSE

The purpose of this chapter is to examine the relationship between ASVAB, the SPSS package, and the QLP 1100 system.

B. INTRODUCTION

ASVAB was designed to generate data quickly. Therefore, ASVAB alone will save the user a considerable amount of time. However, ASVAB has no analytic capability, and only limited data management capabilities. Once ASVAB is implemented, the user may find many applications which use the data in analysis and data management.

The analysis of the ASVAB data may prove to be prohibitive if a computer package is not used. For example, the computation of the mean and standard deviation of a set of composite scores is simple in concept. However in practice, given a sample size of 100 or more, the process becomes extremely time consuming if done manually. Other analysis such as analysis of variance, or regression analysis is almost impossible to do efficiently without computer assistance. Therefore, it is imperative that ASVAB output be linked to the SPSS package.

The analysis done by DOES is not strictly for internal use, but usually arises from external requests. In regard to this, the information provided in the ASVAB output files

may not be sufficient or appropriate to satisfy external requirements. In addition, DOES may be required to submit various reports to the external agencies. A manual process of coding new ASVAB output formats, and typing reports is an inefficient use of time and personnel. This type of data management can be better done using the QLP 1100 system. Once again, the volume of potential, external requests demands that the QLP 1100 system be integrated with ASVAB.

C. SPSS

The user may be required to use various statistical methods on the data generated by ASVAB. SPSS is a comprehensive package that can be used to accomplish this. An SPSS program consists of specific keywords that are used to generate the analysis, as well as the data to be analyzed. The relationship between ASVAB and the SPSS program is shown in Figure 6.

One area for which the user may require SPSS is regression analysis. DOES is responsible for developing selection criteria for attendance to special infantry training. Regression analysis may be used to develop a model for selection criteria. This model will be described in order to provide an example of how ASVAB output can be used with an SPSS program.

In this example, assume a number of soldiers are selected at random to attend a special course. Upon completion of

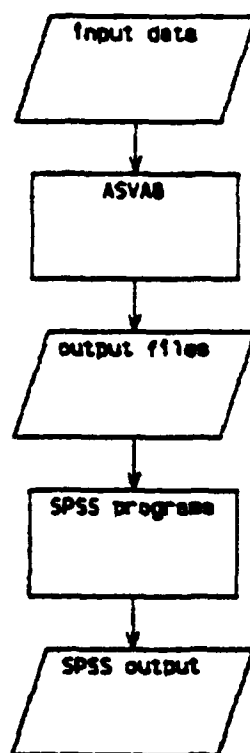


Figure 6. Relationship Between ASVAB and SPSS Programs

the course, the soldiers are administered the appropriate skill qualification test (SQT) to measure their ability to perform the tasks they were trained in. The results of the test are recorded along with the soldier's social security number. Regression analysis is used to develop a model that will predict SQT results as a function of some independent variables. In this example the 10 ACB composite scores are used to predict SQT results.

The user must process the input data cards in ASVAB. The soldier file should be printed in paper copy to check for entry errors. The SQT scores are entered in columns 11 through 15 of File 12, the composite summary. This may be done on disk from a terminal if available, or on the computer cards from the file if interactive processing is not available. The SPSS program is created on file or punched out on computer cards as appropriate. The new data are placed in the appropriate section of the file or deck (see Appendix H). The SPSS program generates all the necessary data to develop the model in final form (see Appendix J).

The R-square value is used to determine the percentage of variability in the SQT scores as explained by the variance in the ten composite scores. A high R-square value will mean that the ten composite scores are good predictors of SQT results. A low R-square value means that the ten composite scores should not be used as predictors. Further analysis such as checking for data points that are outliers, and

residual analysis is done before the model developed by the SPSS package is accepted.

Once the analysis is completed and the model is validated, the SQT model becomes the basis for further selections to the training. SPSS can be used to compute the predicted SQT results for potential attendees prior to their training. In this case, the input cards are processed in ASVAB. Since the soldiers have not attended the training, no SQT results are entered in File 12. An SPSS program is created to compute the predicted SQT results from the model developed earlier (see Appendix I). The SPSS output will contain the predicted SQT results which can be used to select soldiers with the highest potential for success (see Appendix K).

In many instances the selection requirements may be more complex than just predicted SQT scores. For example, the training may involve special infantry combat training. In this case only males with infantry military occupation specialties, and high predicted SQT result can attend. The user would have to manually cross reference the list of attendees based on SQT results, with the soldier file to insure that the additional requirements are met. Alternatively, an application specific computer program can be developed to do the cross referencing. However, the development time may be greater than the time needed to manually cross reference the data. In either case time is not being used effectively.

In general, ASVABMIS does not have the flexibility to provide information on an as-needed basis. In its present

form, ASVABMIS can only provide the information from its output files. A simple request for the number of black soldiers in the data pool requires additional programming, or a manual search.

In addition, ASVABMIS can not generate any reports. Once ASVAB and the SPSS package are used to generate and analyze data, any required reports must be manually created and typed.

In its present form, ASVABMIS is a rigid system unable to process requests for additional information without further programming. However, the QLP 1100 system provides a general program to handle additional requests that does not require additional programming.

D. QLP 1100

QLP 1100 provides a means to create a data base from the ASVAB output files. Since QLP 1100 is a COBOL-based system, the ASVAB output files must be converted to a COBOL format. The QLP 1100 system allows for both the conversion of FORTRAN files to COBOL files, and COBOL files to FORTRAN files.

QLP 1100 is not a true data base management system because many separate files are used, rather than a unique data base, to provide the data for manipulation. However, the end result appears the same. QLP 1100 creates a set of data that can be accessed by a query language.

In order to create the data set two steps are necessary. The first step is to define the physical and logical

structure of the files that will be used. The second step is to define an application for the files by specifying which files will be included, and how the files will be used. Once these steps are completed an environment in which QLP 1100 can operate is established. Finally, the QLP 1100 query language is used to query the files as if they were a data base in order to generate reports and to manipulate data.

Two separate processors and their respective commands are used to accomplish the steps in creating the data base. The file definition processor (FDP) accomplishes the first step by defining the physical and logical attributes of the data files to be accessed, through the use of the FDP source input. More specifically, the FDP defines the file assignment characteristics, the physical file organization specifications, the name and structure specification for all records contained in the file, and the name and attribute specifications for all data items in the records. The FDP produces an internally formatted file specification which is used as input to the application definition processor (ADP).

The ADP accomplishes the second step by specifying which files, as described by the FDP, will be used in the application, as well as how the files will be used through the use of the ADP source input. The ADP produces internally formatted tables which are used by the query language processor (QLP) as input. The QLP then produces the final output as

directed by the user through the use of the query language input. This process is depicted in Figure 7.

1. File Definition Processor Source Input

The FDP syntax contains two major subdivisions. The environment division names a file, and may specify other optional file information as deemed necessary by the user. The data division provides information about the physical structure, identification, and record name pertaining to a file. A simple FDP syntax containing the minimum required language is shown below. The line numbers are provided by the author for line identification, and are not a part of the syntax.

```
01      ENVIRONMENT DIVISION
02          SELECT file-name
03          ASSIGN TO DISK external-file-name
04      DATA DIVISION
05          FD file-name
06          LABEL RECORD IS STANDARD.
```

In this example, lines 01 and 04 specify the two subdivisions. Line 02 selects an existing file from storage by its name file-name. Line 03 allocates a storage medium which can either be disk or tape. The external file name is optional, and is the name under which the file will be stored. If the external file name is not specified, the new file will be stored using the first twelve characters of the selected file

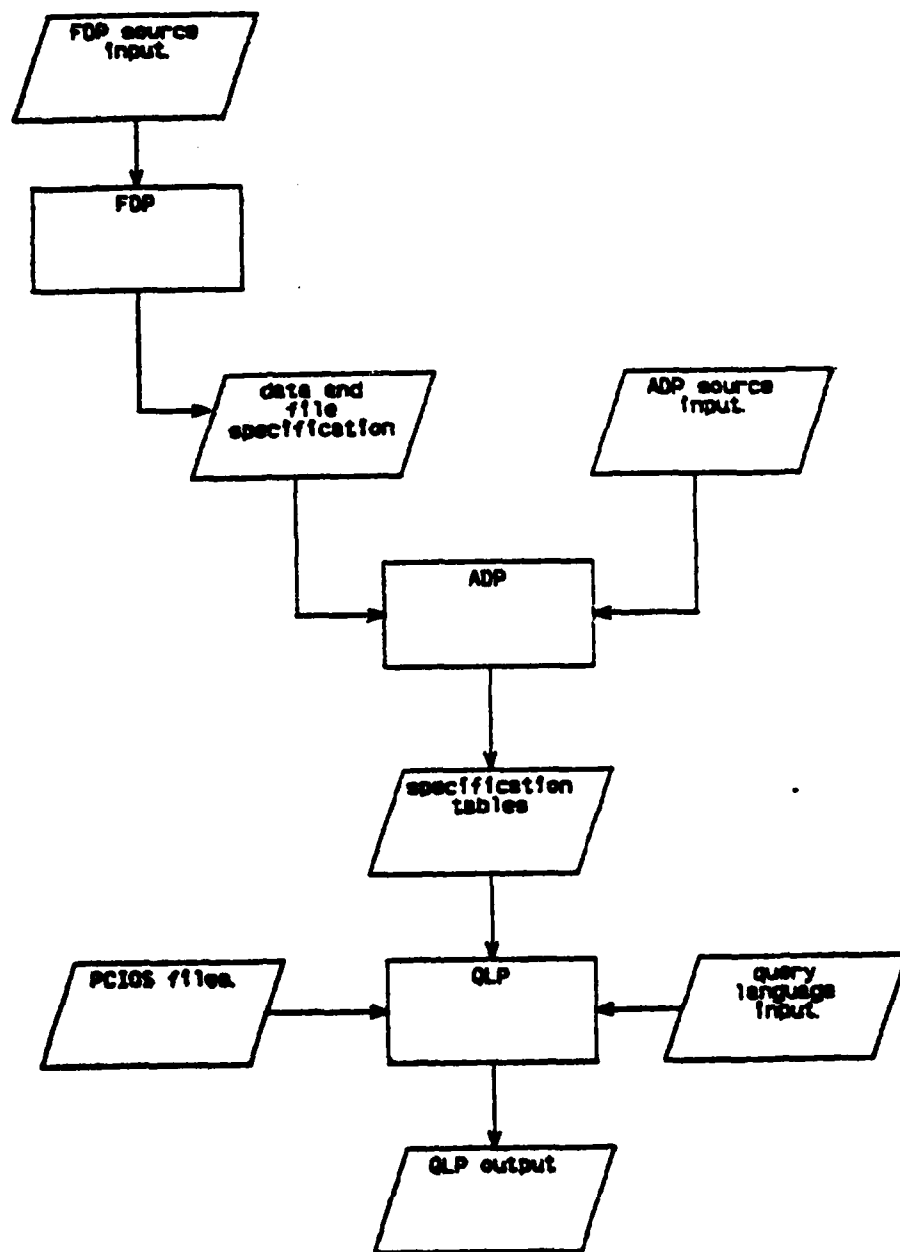


Figure 7. The QLP 1100 System

name. Line 05 is a link between the two divisions. The file name in line 05 must match the file name in line 02. Line 06 specifies that explicit labels exist for the file. The user may also use LABEL RECORD IS OMITTED which specifies that no labels exist for the file. A label is a heading, and a label record is a record that contains file headings.

The FDP syntax may be expanded based on the type of file to be created. If the file is to be a sequential file the syntax may be expanded as shown below.

```
01      ENVIRONMENT DIVISION
02      INPUT-OUTPUT SECTION
03      FILE CONTROL
04          SELECT file-name
05          ASSIGN TO DISK external-file-name
06          FILE QUALIFIER IS qualifier-name
07          ORGANIZATION IS SEQUENTIAL
08          SORT KEY IS ASCENDING data-name
09      DATA DIVISION
11      FD file-name
10          LABEL RECORD IS STANDARD
```

In this example, lines 01, 04, 05, 09, 10 and 11 are unchanged from the previous basic syntax. Lines 02 and 03 are descriptive and add further clarity. Line 06 specifies a file qualifier that is used in the calling sequence when the FDP is invoked. Line 07 specifies that the file is to be

sequentially ordered. Line 08 specifies that the file will be sorted in an ascending order according to the values assigned to the key which is specified by the data name. For example, lines 07 and 08 could be used to sort a file in ascending order according to the social security account number by specifying:

```
ORGANIZATION IS SEQUENTIAL
SORT KEY IS ASCENDING SSAN
```

The user may also specify that the file be sorted in descending order by substituting DESCENDING for ASCENDING in line 08.

Similarly, the basic syntax may be expanded if the file is to be an indexed file rather than a sequential file. If the file is an indexed file the syntax may be expanded as shown below. In either case the user must decide which type of file is the most appropriate.

```
01      ENVIRONMENT DIVISION
02      INPUT-OUTPUT SECTION
03      FILE CONTROL
04          SELECT file-name
05          ASSIGN TO DISK external-file-name
06          FILE QUALIFIER IS qualifier-name
07          ORGANIZATION IS INDEXED
08          RECORD KEY IS data-name
09      DATA DIVISION
10      FD file-name
11          LABEL RECORD IS STANDARD
```

In this example, lines 01, 04, 05, 09, 10 and 11 are unchanged from the basic syntax. Lines 02, 03, and 06 have the same meaning as in the previous sequential example. Line 07 specifies that the file is random access rather than sequential. Line 08 specifies that a record can be identified by its key which has unique values, and is specified by the data name provided by the user. For example, since social security account numbers are unique they can be used to identify any record in a file.

Since the data division specifies the physical structure, identification, and record name of a file, the data division may be further specified with optional syntax if required. However, this syntax is normally used when default specifications are not appropriate. Some of the available optional syntax language is described below.

BLOCK CONTAINS integer-1 TO integer-2 RECORDS

This syntax specifies the block size in terms of the range of record numbers in the block.

DATA RECORD IS record-name-1

DATA RECORDS ARE record-name-1, record-name-2

This syntax is used when there is more than one record type in the file, and specifies the record selection by its name. More than one record type may be selected in multiple type files by listing the appropriate record names.

level number record-name

level number data-name PICTURE IS character string

This syntax further describes the selected data record. The level number specifies the hierarchy of the data. For example, a record has a higher hierarchy than a data element, and would therefore have a level number of 01, while all the data elements would have level numbers of 02.

The PICTURE syntax describes the general characteristics of the data element as defined by the character string. It is in effect a format statement. The character string is a combination of symbols comprised of A, S, V, X and 9. These symbols are shown in Table 14.

TABLE 14
PICTURE CHARACTERS

<u>CHARACTER</u>	<u>DESCRIPTION</u>
A	A character position which can only contain a letter of the alphabet or a space.
S	A character position used to denote the presence of an operational sign, i.e., plus or minus.
V	A character position used to denote the location of a decimal point.
X	A character position which contains any allowable character from the computer's character set, i.e., \$, /.
9	A character representing a numeral.

An example of the picture clause is shown below.

```
01  RA
    02  RA1  PICTURE IS  9999
    02  RA2  PICTURE IS  S9V999
    02  RA3  PICTURE IS  AAAA
    02  RA4  PICTURE IS  A999
```

This example specifies that the record RA has the highest hierarchy. The record has four data elements specified as RA1, RA2, RA3 and RA4. Furthermore, the syntax specifies that RA1 is a numeric character made up of four numerals. This specification may also be shown in a shorter notation by using a 9(4) character string. RA2 is a signed numeric character having a decimal point following the first numeral, and three numeral decimal values. RA3 is an alphabetic character with four positions. It may optionally be specified by an A(4) character string. Finally, RA4 is an alphanumeric character specifying one alphabetical character followed by three numerals. Its optional specification could have been shown as A9(3).

Combining the previous examples, a complete FDP source input is shown below.

ENVIRONMENT DIVISION

INPUT-OUTPUT SECTION

FILE CONTROL

SELECT file-name

ASSIGN TO DISK external-file-name

```

FILE QUALIFIER IS qualifier-name
ORGANIZATION IS INDEXED
RECORD KEY IS data-name-2

DATA DIVISION

FD file-name

    LABEL RECORD IS OMITTED

    BLOCK CONTAINS integer-1 to integer-2 RECORDS

    DATA RECORD IS record-name

01 record-name

    02 data-name-1 PICTURE IS AAA

    02 data-name-2 PICTURE IS 999

```

In this example, the syntax specifies that a file called file-name will be taken from storage and assigned to disk storage under the name external-file-name. The file is assigned a qualifier name which is used in the FDP calling sequence when invoking the FDP. The file is an indexed file, and each record in the file can be identified by the key named data-name-2.

The data division further specifies that the file has no explicit labels. In addition, the range of record numbers in a block is specified. Also, the file contains multiple record types, but only one record type having the name record name will be used. Finally, the record has two data elements, the first of which is a three position alphabetic character. The last data element is specified to be a three numeral numeric character.

When invoked, the FDP will accept the FDP source input and initiate a completely independent run. That is, different constructs of FDP source input will produce different FDP run results. The output from a FDP run consists of an internally formatted description of the file specification defined by the FDP source input. This output, along with the ADP source input, is used as input into the ADP.

2. Application Definition Processor Source Input

The ADP links the FDP output and the separate files to produce a processor common input/output system (PCIOS) file. The PCIOC file becomes the data base to be accessed by the QLP. This is accomplished through the ADP source input.

The ADP syntax has two divisions. The first division is the identification division which simply contains the name of the application. The second division is the data division which contains the names of the files to be used in the application, as well as specifying the location of the FDP file description, and the usage restriction on the file. A complete ADP source input syntax is shown below.

```
01      IDENTIFICATION DIVISION
02          APPLICATION NAME IS application-name
03      DATA DIVISION
04          FILE SECTION
05              FILE NAME IS file-name-1
06              DEFINED IN FILE file-name-2
```

07	ELEMENT element-name-1
08	QUALIFIER qualifier-name-1
09	USE IS RETRIEVAL

In this example, lines 01, 03 and 04 are descriptive and provide clarity. Line 02 assigns a name to this specific application. Different applications can be applied to the same FDP run. Therefore, the application name is used to identify specific runs. The application name is also used to define where the application run will reside. This is important when linking to the QLP. Line 05 specifies which files will be used in this application and is the same file specified in the SELECT clause, and the FD clause in the FDP source input. Lines 06, 07, and 08 specify precisely where the FDP run output is located. This information is included in the FDP calling sequence. Finally, line 09 defines how the application will be used. The word RETRIEVAL means that the file can only be read. The words UPDATE or OUTPUT may be substituted for RETRIEVAL. The word OUTPUT specifies that the file is intended for initial creation. Finally, the word UPDATE specifies that the file may be both read and written.

The ADP source input along with the FDP output produces an independent ADP run. The output from the ADP run produces internally formatted tables that are used as input into the QLP.

3. FDP and ADP Call Commands

Before the FDP or ADP source input can be used to produce processor runs, each processor must be invoked using call commands. The basic syntax for the call commands is shown below and are described in Table 15.

@ file-name. FDP, options, SI, RO, SO

@ file-name. ADP, options, SI, RO, SO

TABLE 15

CALL SYNTAX

<u>ELEMENT</u>	<u>DESCRIPTION</u>
file-name	Is the name of the file containing the absolute FDP or ADP.
options	Are specified through the choice of one or more option letters.
D	Produce an allocation summary for each record in the file. This may be used only on the ADP.
I	Insert a new symbolic element in the program file.
P	Card image input, if any, is in fieldata. The output symbolic element is in fieldata.
Q	Output symbolic element is in ACIII. Card image input, if any, is in ASCII.
S	The source input will be echoed to the user.
U	Update and produce a new cycle of the symbolic element.
W	List correction lines.

The characters SI, RO, and SO are parameters that stand for source input, relocatable output, and source output, respectively. The meaning of the parameters are shown in Table 16.

TABLE 16
PARAMETER DESCRIPTIONS

<u>PARAMETER</u>	<u>DESCRIPTION</u>
SI	If a new element is being introduced from the run, this parameter specifies the file into which the new element is placed and the name it is given.
RO	This parameter specifies the name and program file into which the element produced by the processor is placed.
SO	This parameter specifies the name and the file for the updated symbolic element produced.

An example of a processor call is shown below.

```
@ FDP.FDP,S FDP*FDPEXAMPLE.SOURCE,FDP*FDPEXAMPLE.RELOCATABLE
```

In this example, the file name is FDP, and the option is S. The source input contains the actual source description for this file. More specifically the qualifier is FDP, the file name is FDPEXAMPLE, and the element is RELOCATABLE. The relocatable output is specified by the statement FDP*FDPEXAMPLE.RELOCATABLE and is the location where the FDP output will reside.

The processor calls are combined with the FDP and ADP source input to completely define the file and application

definitions. This combination creates a PCIOS file which can be accessed by the QLP.

The procedure for creating the PCIOS file from the 3 ASVAB summary files raw, standard, and composite is explained below.

STEP 1: Produce an FDP description of file raw.

@FDP.FDP,S FDP*SCORES.SOURCE,FDP*SCORES.RELOC-RAW

ENVIRONMENT DIVISION

INPUT-OUTPUT SECTION

FILE-CONTROL

SELECT RAW

ASSIGN TO DISK

FILE QUALIFIER IS FDP

ORGANIZATION IS INDEXED

RECORD KEY IS SSAN

DATA DIVISION

FILE SECTION

FD RAW

LABEL RECORDS ARE STANDARD

STEP 2: Produce an FDP description of file standard.

@FDP.FDP,S FDP*SCORES.SOURCE,FDP*SCORES.RELOC-STAND

ENVIRONMENT DIVISION

INPUT-OUTPUT SECTION

FILE-CONTROL

SELECT STANDARD
ASSIGN TO DISK
FILE QUALIFIER IS FDP
ORGANIZATION IS INDEXED
RECORD KEY IS SSAN
DATA DIVISION
FILE SECTION
FD STANDARD
LABEL RECORDS ARE STANDARD

STEP 3: Produce an FDP description of the file composite.

@FDP.FDP,S FDP*SCORES.SOURCE,FDP*SCORES.RELOC-COMP

ENVIRONMENT DIVISION
INPUT-OUTPUT SECTION
FILE CONTROL
SELECT COMPOSITE
ASSIGN TO DISK
ORGANIZATION IS INDEXED
RECORD KEY IS SSAN
DATA DIVISION
FILE SECTION
FD COMPOSITE
LABEL RECORDS ARE STANDARD

STEP 4: The ADP is used to identify which files are to
be used in this application.

@ADP.ADP,ADP*SCORES.SOURCE-BATTERY,ADP*SCORES,ACB-BATT

IDENTIFICATION DIVISION

APPLICATION NAME IS ACB-BATT

DATA DIVISION

FILE SECTION

FILE NAME IS RAW

DEFINED IN FILE SCORES

ELEMENT RELOC-RAW

QUALIFIER IS FDP

USE IS RETRIEVAL

FILE NAME IS STANDARD

DEFINED IN FILE SCORES

ELEMENT RELOC-STAND

QUALIFIER IS FDP

USE IS RETRIEVAL

FILE NAME IS COMPOSITE

DEFINED IN FILE SCORES

ELEMENT RELOC-COMP

QUALIFIER IS FDP

USE IS RETRIEVAL

STEP 5: The QLP is invoked

INVOKE ACB-BATT OF ADP*SCORES

With this accomplished all files and applications are defined. The application tables produced by the ADP are saved in ADP*SCORES.ACB-BATT, and the QLP has been invoked and is prepared to accept QLP 1100 query commands.

4. Query Language Processor, 1100 Series

The QLP accepts as input the ADP output and its specific QLP 1100 language. The QLP output consists of reports and data files that can be stored or printed. The QLP 1100 language is subdivided into eight functional facilities. Each facility has several commands associated with it. The facilities are:

Data system interface facilities

Conversational facilities

Data selection facilities

Operational facilities

Definitional facilities

Logic control and manipulative facilities

Data access and data base network facilities

Savefile facilities

The last four facilities are complex and are intended for use by experienced programmers. They will be given only a cursory explanation. The first four facilities will be explained in more detail.

a. Data System Interface Facilities

(1) INVOKE. INVOKE establishes a link between the user and the data files to be accessed. The syntax is:

INVOKE application-name OF qualifier-name*file-name

An example of the syntax is:

INVOKE ACB-BATT OF ADP*SCORES

This command links the user to application ACB-BATT and the stored output file ADP*SCORES.

(2) EXIT. EXIT releases the link between the user and the file, and terminates QLP processing.

b. Conversational Facilities

(1) LIST. LIST provides the user with a list of data values as specified by the WHERE clause. The syntax is:

LIST data-identifier WHERE expression

An example of the syntax is:

LIST SSAN WHERE AFQT > 90

This command will provide the response:

SSAN = 242968813

SSAN = 016558905

The LIST command may be expanded according to the user's needs. An example of an expanded syntax is:

LIST SSAN, MOS WHERE AFQT > 90

This command will provide the response:

SSAN = 242968813

MOS = 11B10

SSAN = 016558905

MOS = 11C30

The SORT command may be used in conjunction with the LIST command in the following manner:

```
LIST SSAN,MOS,AFQT SORTED ON ASCENDING AFQT  
WHERE AFQT > 90
```

This command will provide the response:

SSAN = 016558905

MOS = 11C30

AFQT = 96

SSAN = 242968813

MOS = 11B10

AFQT = 92

(2) COUNT. COUNT tallies the number of occurrences specified by the COUNT clause, and qualified by the WHERE clause. The syntax is:

COUNT data-identifier WHERE expression

An example of the syntax is:

```
COUNT SSAN WHERE AFQT < 89
```

This command will provide the response:

312 SSAN RECORDS SELECTED

(3) CHANGE. CHANGE modifies, as specified, the data elements qualified by the WHERE clause. The syntax is:

CHANGE data-identifier=expression WHERE expression

An example of the syntax is:

CHANGE DES='AAA' WHERE SSAN = 041358991

This command changes the DES value to AAA for the record identified by the social security number 041358991.

(4) REPEAT. REPEAT executes the previous command with a new WHERE clause. The syntax is:

REPEAT WHERE expression

The REPEAT command may be used when a previous command did not provide valid results. For example, if the syntax LIST SSAN WHERE AFQT > 101 did not provide valid results because there is no such percentile, the list command may be repeated by commanding:

REPEAT WHERE AFQT > 75

c. Data Selection Facilities

WHERE specifies the criteria for data selection.

The syntax is:

WHERE expression

d. Operational Facilities

OUTPUT defines the destination of subsequent query output to a terminal, printer, or file. The syntax is:

OUTPUT TO device

For example, in order to send a list command to the printer and then return all output to the originating device the following commands are issued:

OUTPUT TO PRINTER

LIST SSAN WHERE AFQT > 70

OUTPUT TO ORIGINATOR

e. Definitional Facilities

FORMAT allows the user to specify a simple tabular output form with column headings. The syntax is:

FORMAT format-name

format-clause

END FORMAT

The FORMAT syntax is used with other syntax to generate the tables. For example, in order to generate a table with headings, the LIST command is used as shown below.

FORMAT FORM

'SSAN' SSAN 1, 'PERCENTILE' AFQT 20

END FORMAT

LIST USING FORMAT FORM WHERE AFQT > 90

The FORMAT syntax specifies that there are two columns to be printed. The first column occupies print positions 1 through 19, while the second column starts in column 20. The table will appear as shown below.

SSAN	PERCENTILE
242968813	92
016558905	96

f. Logic Control and Manipulative Facilities

These facilities allow the user to create an algorithm of commands, known as a macro, that can accomplish an application that can not be accomplished with one command. The commands that accomplish this are DO, IF, LEAVE, and RETURN.

g. Data Access and Data Base Network Facilities

These facilities allow the user to process records one at a time, by specifying pointers to specific records.

h. SAVEFILE Facilities

SAVE allows the user to save files. The syntax is:

SAVE savefile-name INTO PERMANENT SAVEFILE.

In order to save a file named FILE1 the following command is issued:

SAVE FILE1 INTO PERMANENT SAVEFILE

5. Summary Example

In this example the user has a requirement to provide a selection listing of potential attendees for a special school. The selection listing must provide the social security numbers of the attendees selected, along with a justification for the selection. In addition the number of potential attendees who were selected must be provided.

The user determines that the regression model developed from previous classes will be used to predict SQT scores. The selection criteria are determined from predicted SQT scores first, and AFQT percentile second. The user also determines that only potential attendees with a predicted SQT score of 50 or better are qualified to attend.

The user processes the input data cards in ASVAB, and uses the SPSS program to compute the predicted SQT result. The data in the composite score summary file, and the predicted SQT results are combined into one file using the QLP 1100 command;

BUILD COBOL FILE REGDATA ON DISK FROM SSAN, SQT, AFQT,
CO, FA, MM, GM, CL GT, EL, SC, ST, OF

The user must select the file REGDATA in the FDP and ADP source input. The QLP is then invoked. In order

to satisfy the requirement the user would issue the QLP
1100 commands shown below.

FORMAT REQUIRES

'SSAN'SSAN1, 'SQT'/'RANK' SQT12, 'AFQT'/'RANK' AFQT16

END FORMAT

OUTPUT TO PRINTER

LIST USING FORMAT REQUIRES SORTED ON DESCENDING SQT

SORTED ON DESCENDING AFQT WHERE SQT > 50

COUNT SSAN WHERE SQT > 50

OUTPUT TO ORIGINATOR

EXIT

This QLP 1100 query session will produce the output shown
below.

SSAN	SQT	AFQT
	RANK	RANK
232489961	85	60
016558843	75	55
114863105	75	50
⋮		
404358221	51	45

97 SSAN RECORDS SELECTED

The entire ASVABMIS system as used in the above
example is shown in Figure 8. This figure shows that ASVAB
can be integrated with SPSS and QLP 1100 into one information
system.

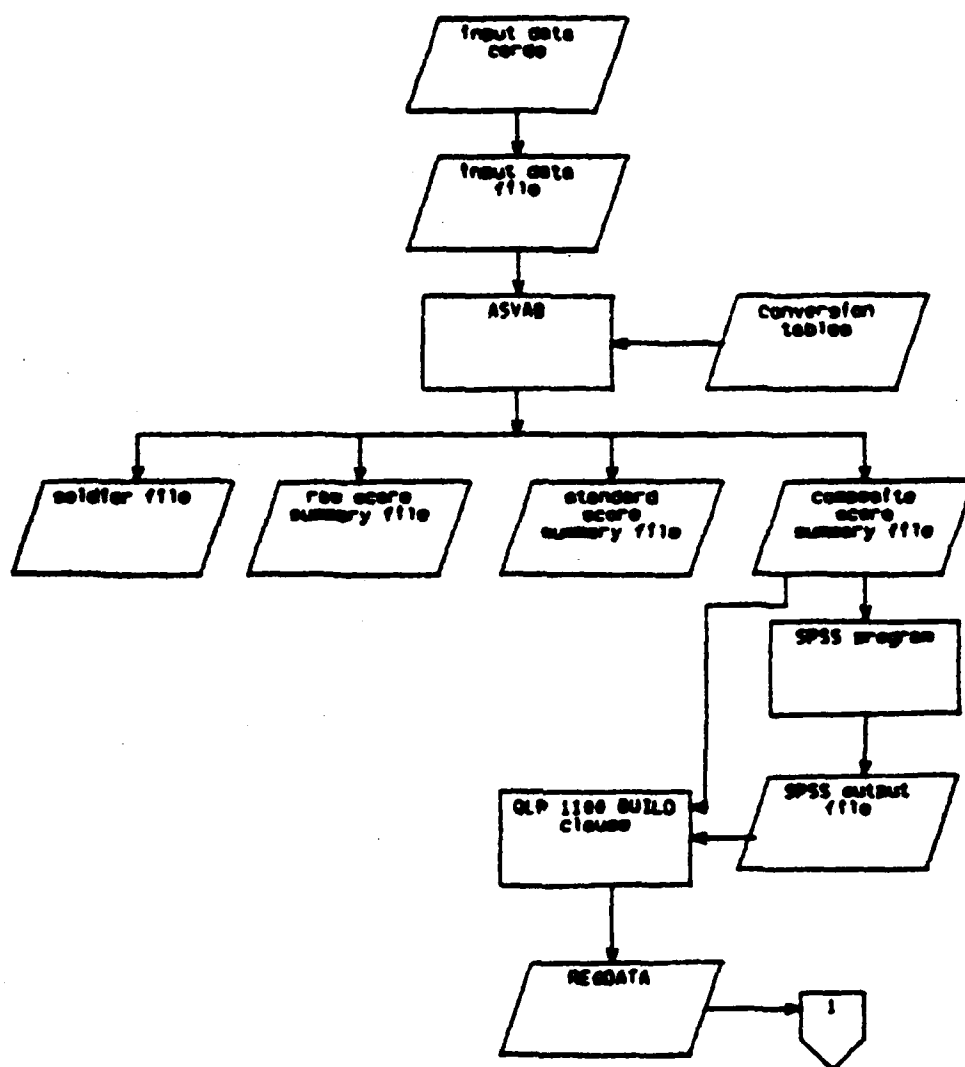


Figure 8. The ASVABMIS System

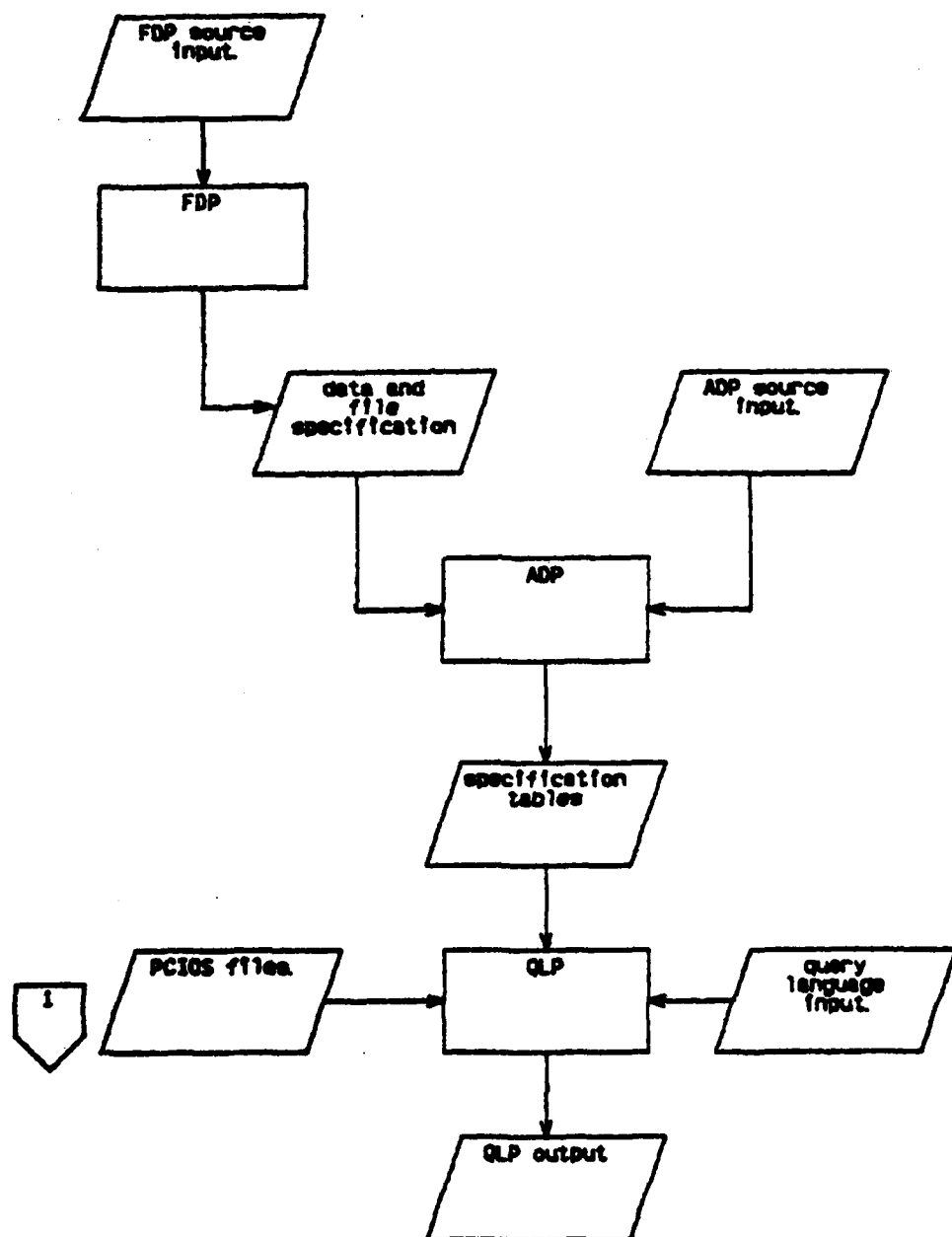


Figure 8. (CONT.)

V. CONCLUSIONS

The management information system has been developed in order to provide the user with a means to generate, analyze and manage data concerning the Armed Services Aptitude Battery. The ASVAB program is complete and can be immediately installed by the user. This thesis provides the necessary program documentation, and the user need not acquire additional documents. The FORTRAN output files can be used immediately with SPSS programs. However, since only a few sample SPSS programs are provided, the user may find it necessary to obtain the SPSS manuals listed in the bibliography in order to develop other programs. The sample SPSS programs were processed on a computer and are valid. The user can install them as they are written.

The treatment of the QLP 1100 system was introductory only. The user will find it necessary to obtain QLP 1100 manuals in order to further develop this part of the system. The chapter covering the QLP 1100 system could not provide all the necessary information for a complete understanding of QLP 1100. However, the coverage of QLP 1100 is specific enough to demonstrate the ease of operation, and efficiency of the system. This in itself should prompt further development of the QLP 1100 system.

The ASVABMIS system has been designed for use by U.S. Army agencies. The ASVAB program can be expanded for use

by the Air Force and Marine Corps. The same raw scores are used by all 3 services. The major difference is the type of composites computed.

The ASVABMIS system was developed specifically for the Infantry Center's Department of Evaluation and Standardization. The applications discussed in this thesis are relevant to similar departments of the Armor, Field Artillery, and other centers.

A future expansion of the system worthy of note is the installation of ASVABMIS on a mini or micro computer. The ASVAB program can be written in BASIC. In addition, vendor-developed statistical packages and data base systems can be used on small computers in a manner similar to SPSS and QLP 1100.

In conclusion, the ASVABMIS system developed in this thesis will provide the user with the ability to generate and analyze ASVAB data. A limited ability to manage data is also provided. However, extensive data management will require further development by the user.

APPENDIX A
INPUT DATA FILE

272558843640420128202160301118103547152003120617201009051312	AKL830227
01653224163112511810920060111810368713402522151612231009171408121322	AAA821109
21386552763052123821007070254E1040922422301445792520241948	AAA831015
23544232163052311821113060111810418522321291344722020211823	AAA831110
01534223764080911810925060111810377515392521151612231112171310121332	AKL840915

ASVAB 5/6/7 SSS CONVERSION TABLE

84

APPENDIX C

ASVAB 8/9/10 SSS CONVERSION TABLE

[illegible]

	GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
58						59					
59						60					
60						60					
61						61					
62						62					
63						62					
64						63					
65						64					
66						64					
67						65					
68						65					
69						66					
70						67					
71						67					
72						68					
73						69					
74						69					
75						70					
76						70					
77						71					
78						72					
79						72					
80						73					
81						73					
82						74					
83						75					
84						75					

APPENDIX D

ASVAB 5 COMPOSITE CONVERSION TABLE

SSS	GT	GM	EL	CL	MM	SC	CO	FA	OF	ST
1	41	41	41	41	41	41	41	41	41	41
2	41	41	41	41	41	41	41	41	41	41
3	41	41	41	41	41	41	41	41	41	41
4	41	41	41	41	41	41	41	41	41	41
5	41	41	41	41	41	41	41	41	41	41
6	41	41	41	41	41	41	41	41	41	41
7	41	41	41	41	41	41	41	41	41	41
8	41	41	41	41	41	41	41	41	41	41
9	41	41	41	41	41	41	41	41	41	41
10	41	41	41	41	41	41	41	41	41	41
11	41	41	41	41	41	41	41	41	41	41
12	41	41	41	41	41	41	41	41	41	41
13	41	41	41	41	41	41	41	41	41	41
14	41	41	41	41	41	41	41	41	41	41
15	41	41	41	41	41	41	41	41	41	41
16	41	41	41	41	41	41	41	41	41	41
17	41	41	41	41	41	41	41	41	41	41
18	41	41	41	41	41	41	41	41	41	41
19	41	41	41	41	41	41	41	41	41	41
20	41	41	41	41	41	41	41	41	41	41
21	41	41	41	41	41	41	41	41	41	41
22	41	41	41	41	41	41	41	41	41	41
23	41	41	41	41	41	41	41	41	41	41
24	41	41	41	41	41	41	41	41	41	41
25	41	41	41	41	41	41	41	41	41	41
26	41	41	41	41	41	41	41	41	41	41
27	41	41	41	41	41	41	41	41	41	41
28	41	41	41	41	41	41	41	41	41	41
29	41	41	41	41	41	41	41	41	41	41
30	41	41	41	41	41	41	41	41	41	41
31	41	41	41	41	41	41	41	41	41	41
32	41	41	41	41	41	41	41	41	41	41
33	41	41	41	41	41	41	41	41	41	41
34	41	41	41	41	41	41	41	41	41	41
35	41	41	41	41	41	41	41	41	41	41
36	41	41	41	41	41	41	41	41	41	41
37	41	41	41	41	41	41	41	41	41	41
38	41	41	41	41	41	41	41	41	41	41
39	41	41	41	41	41	41	41	41	41	41
40	41	41	41	41	41	41	41	41	41	41
41	41	41	41	41	41	41	41	41	41	41
42	41	41	41	41	41	41	41	41	41	41
43	41	41	41	41	41	41	41	41	41	41
44	41	41	41	41	41	41	41	41	41	41
45	41	41	41	41	41	41	41	41	41	41
46	41	41	41	41	41	41	41	41	41	41
47	41	41	41	41	41	41	41	41	41	41
48	41	41	41	41	41	41	41	41	41	41
49	41	41	41	41	41	41	41	41	41	41
50	41	41	41	41	41	41	41	41	41	41
51	41	41	41	41	41	41	41	41	41	41
52	41	41	41	41	41	41	41	41	41	41
53	41	41	41	41	41	41	41	41	41	41
54	41	41	41	41	41	41	41	41	41	41
55	41	41	41	41	41	41	41	41	41	41
56	41	41	41	41	41	41	41	41	41	41
57	41	41	41	41	41	41	41	41	41	41
58	41	41	41	41	41	41	41	41	41	41

88

SSS	125	GT	GM	EL	CL	MM	SC	CO	FA	OF	ST
126	127	128	129	130	131	132	133	134	135	136	137
138	139	140	141	142	143	144	145	146	147	148	149
150	151	152	153	154	155	156	157	158	159	160	161
162	163	164	165	166	167	168	169	170	171	172	173
174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197
198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232	233
234	235	236	237	238	239	240	241	242	243	244	245
246	247	248	249	250	251	252	253	254	255	256	257
258	259	260	261	262	263	264	265	266	267	268	269
270	271	272	273	274	275	276	277	278	279	280	281
282	283	284	285	286	287	288	289	290	291	292	293
294	295	296	297	298	299	300	301	302	303	304	305
306	307	308	309	310	311	312	313	314	315	316	317
318	319	320	321	322	323	324	325	326	327	328	329
330	331	332	333	334	335	336	337	338	339	340	341
342	343	344	345	346	347	348	349	350	351	352	353
354	355	356	357	358	359	360	361	362	363	364	365
366	367	368	369	370	371	372	373	374	375	376	377
378	379	380	381	382	383	384	385	386	387	388	389
390	391	392	393	394	395	396	397	398	399	400	401
402	403	404	405	406	407	408	409	410	411	412	413
414	415	416	417	418	419	420	421	422	423	424	425
426	427	428	429	430	431	432	433	434	435	436	437
438	439	440	441	442	443	444	445	446	447	448	449
450	451	452	453	454	455	456	457	458	459	460	461
462	463	464	465	466	467	468	469	470	471	472	473
474	475	476	477	478	479	480	481	482	483	484	485
486	487	488	489	490	491	492	493	494	495	496	497
498	499	500	501	502	503	504	505	506	507	508	509
510	511	512	513	514	515	516	517	518	519	520	521
522	523	524	525	526	527	528	529	530	531	532	533
534	535	536	537	538	539	540	541	542	543	544	545
546	547	548	549	550	551	552	553	554	555	556	557
558	559	560	561	562	563	564	565	566	567	568	569
570	571	572	573	574	575	576	577	578	579	580	581
582	583	584	585	586	587	588	589	590	591	592	593
594	595	596	597	598	599	600	601	602	603	604	605
606	607	608	609	610	611	612	613	614	615	616	617
618	619	620	621	622	623	624	625	626	627	628	629
630	631	632	633	634	635	636	637	638	639	640	641
642	643	644	645	646	647	648	649	650	651	652	653
654	655	656	657	658	659	660	661	662	663	664	665
666	667	668	669	670	671	672	673	674	675	676	677
678	679	680	681	682	683	684	685	686	687	688	689
690	691	692	693	694	695	696	697	698	699	700	701
702	703	704	705	706	707	708	709	710	711	712	713
714	715	716	717	718	719	720	721	722	723	724	725
726	727	728	729	730	731	732	733	734	735	736	737
738	739	740	741	742	743	744	745	746	747	748	749
750	751	752	753	754	755	756	757	758	759	760	761
762	763	764	765	766	767	768	769	770	771	772	773
774	775	776	777	778	779	780	781	782	783	784	785
786	787	788	789	790	791	792	793	794	795	796	797
798	799	800	801	802	803	804	805	806	807	808	809
810	811	812	813	814	815	816	817	818	819	820	821
822	823	824	825	826	827	828	829	830	831	832	833
834	835	836	837	838	839	840	841	842	843	844	845
846	847	848	849	850	851	852	853	854	855	856	857
858	859	860	861	862	863	864	865	866	867	868	869
870	871	872	873	874	875	876	877	878	879	880	881
882	883	884	885	886	887	888	889	890	891	892	893
894	895	896	897	898	899	900	901	902	903	904	905
906	907	908	909	910	911	912	913	914	915	916	917
918	919	920	921	922	923	924	925	926	927	928	929
930	931	932	933	934	935	936	937	938	939	940	941
942	943	944	945	946	947	948	949	950	951	952	953
954	955	956	957	958	959	960	961	962	963	964	965
966	967	968	969	970	971	972	973	974	975	976	977
978	979	980	981	982	983	984	985	986	987	988	989
990	991	992	993	994	995	996	997	998	999	1000	1001
1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013
1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025
1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037
1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049
1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061
1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073
1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085
1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097
1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109
1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121
1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133
1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145
1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157
1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169
1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181
1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193
1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205
1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217
1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229
1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241
1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253
1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265
1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277
1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289
1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301
1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313
1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325
1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337
1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349
1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361
1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373
1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385
1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397
1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409
1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421
1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433
1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445
1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457
1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469
1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481
1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493
1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505
1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517
1518	1519	1520	1521								

90

ASVAB 6/7 COMPOSITE CONVERSION TABLE

[illegible]

RAM	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

ASVAB 8/9/10 COMPOSITE CONVERSION TABLE

[illegible]

95

96

97

98

APPENDIX G

ASVAB PROGRAM OUTPUT FILES

SSAN	OF BIRTH	(YY/MM/DD)	:	272598843
DATE			:	64/ 4/20
SEX			:	1 - MALE
RACE			:	2 - BLACK
DATE OF ENTRY	(YY/MM/DD)	:	82/ 2/16	
HIGHEST YEAR OF EDUCATION		:	1	
ENTRY PAY GRADE		:	11810	
MOS		:	47	
35 TEST FORM		:	ASVAB 5	
AFQT PERCENTILE		:	47	
SEPARATION PROGRAM DESIGNATOR	:	AKL		
DATE OF SEPARATION	(YY/MM/DD)	:	83/ 2/27	

GI	NO	AC	WK	AR	SP	MK	EI	MC	GS	SI	AI
15	20	3	12	6	17	20	10	9	5	13	12

SUBTEST STANDARD SCORES											
GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI
66	39	21	39	36	59	67	34	46	36	48	50

ARMY CLASSIFICATION BATTERY COMPOSITES											
CO	FA	MM	GM	CL	GT	EL	SC	ST	OF		
71	102	99	79	49	71	76	89	94	116		

SSAN OF BIRTH(VV/MM/DD) : 016532241
 DATE : 63/11/25
 SEX : 1 - MALE
 RACE : 1 - WHITE
 DATE OF ENTRY(VV/MM/DD) : 81/ 9/20
 HIGHEST YEAR OF EDUCATION : HIGH SCHOOL DIPLOMA - 6
 ENTRY PAY GRADE : E1
 MOS : 11B10
 TEST FORM : ASVAB 6
 36 IT PERCENTILE : 87
 AFQT PERCENTILE : 87
 SEPARATION PROGRAM DESIGNATOR : AAA
 DATE OF SEPARATION(VV/MM/DD) : 82/11/ 9

SUBTEST RAW SCORES															
GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI	CM	CA	CE	CC
13	40	25	22	15	16	12	23	10	9	17	14	8	12	13	22

SUBTEST STANDARD SCORES															
GI	NO	AD	WK	AR	SP	MK	EI	MC	GS	SI	AI	CM	CA	CE	CC
60	59	77	53	55	57	51	56	48	45	58	55	44	58	60	63

ARMY CLASSIFICATION BATTERY COMPOSITES					
CO	F	MM	GM	CL	ST
125	115	106	103	122	109
				EL	SC
				114	108
				104	115
				OF	

SSAN	OF BIRTH(Y/M/DD)	:	213065527
DATE		:	63/ 5/21
SEX		:	2 - FEMALE
RACE		:	3 - OTHER
DATE OF ENTRY(Y/M/DD)		:	82/10/ 7
HIGHEST YEAR OF EDUCATION		:	1 YEAR OF COLLEGE - 7
ENTRY PAY GRADE		:	E2 2
MOS		:	54E10
40 TEST FORM		:	ASVAB 10
AFQT PERCENTILE		:	92
SEPARATION PROGRAM DESIGNATOR		:	AAA
DATE OF SEPARATION(Y/M/DD)		:	83/10/15

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
24	22	30	14	45	79	25	20	24	19	48

GS	AR	WK	PC	NO	CS	AS	MK	MC	EI	VE
65	56	56	60	59	72	65	63	65	65	61

CO	138	128	132	133	117	124	144	128	135
ARMY	FA	MM	GM	CL	GT	EL	SC	ST	OF

ERRCR IN RECORD NUMBER: 4 WITH SSAN: 235442321
 ERROR: TEST FORM INCORRECT, COLUMNS 33-34

SSAN	OF BIRTH(YV/MM/DD)	: 235442321
DATE		: 63/ 5/23
SEX		: 1 - MALE
RACE		: 1 - WHITE
DATE OF ENTRY(YV/MM/DD)		: 82/11/13
HIGHEST YEAR OF EDUCATION		: HIGH SCHOOL DIPLOMA - 6
ENTRY PAY GRADE		: E1
HOS		: 1
41 TEST FORM		: 11810
AFCT PERCENTILE		: ASVAB 10
SEPARATION PROGRAM DESIGNATOR: AAA		: 85
DATE OF SEPARATION(YV/MM/DD)		: 83/11/10

GS	AR	WK	PC	NO	CS	AS	MC	EI	VE
23	21	29	13	44	72	20	21	18	23

ERROR IN RECORD NUMBER : 5 WITH SSAN: 015342237
 ERROR: CC INCORRECT, COLUMNS 67-68

SSAN : 015342237
 DATE OF BIRTH(Y Y/M M/D D) : 64/ 8/ 9
 SEX : 1 - MALE
 RACE : 1 - WHITE
 DATE OF ENTRY(Y Y/M M/D D) : 81/ 9/25
 HIGHEST YEAR OF EDUCATION : HIGH SCHOOL DIPLOMA - 6
 ENTRY PAY GRADE : E1
 MOS : 11B10
 37 TEST FORM : ASVAB 7
 AFQT PERCENTILE : 75
 SEPARATION PROGRAM DESIGNATOR: AKL
 DATE OF SEPARATION(Y Y/M M/D D) : 84/ 9/15

GI NO AD WK AR SP MK EI MC GS SI AI CM CA CE CC
 15 39 25 21 15 16 12 23 11 12 17 13 10 12 13 32

SSAN 272598843
 016532241
 213865527
 SUBTEST RAW SCORES
 GI 13 15 20 23 24
 NC 13 13 22 22 22
 AD 13 13 22 22 22
 WK 13 13 22 22 22
 AR 13 13 22 22 22
 SP 13 13 22 22 22
 MK 13 13 22 22 22
 EI 13 13 22 22 22
 MC 13 13 22 22 22
 GS 13 13 22 22 22
 SI 13 13 22 22 22
 AI 13 13 22 22 22
 CM 13 13 22 22 22
 CA 13 13 22 22 22
 CE 13 13 22 22 22
 CC 13 13 22 22 22
 PC 13 13 22 22 22
 CS 13 13 22 22 22
 AS 13 13 22 22 22
 VE 13 13 22 22 22

SSAN 272598843
 016532241
 213865527
 SUBTEST STANDARD SCORES
 GI 66 66 66 66 66
 NC 66 66 66 66 66
 AD 66 66 66 66 66
 WK 66 66 66 66 66
 AR 66 66 66 66 66
 SP 66 66 66 66 66
 MK 66 66 66 66 66
 EI 66 66 66 66 66
 MC 66 66 66 66 66
 GS 66 66 66 66 66
 SI 66 66 66 66 66
 AI 66 66 66 66 66
 CM 66 66 66 66 66
 CA 66 66 66 66 66
 CE 66 66 66 66 66
 CC 66 66 66 66 66
 PC 66 66 66 66 66
 CS 66 66 66 66 66
 AS 66 66 66 66 66
 VE 66 66 66 66 66

ARMY CLASSIFICATION BATTERY COMPOSITES
 SSAN 272598843
 016532241
 213865527
 AFQT 47 87 92
 CO 71 125 138
 FA 102 115 128
 MM 99 106 138
 GM 79 103 132
 CL 49 122 133
 GI 71 109 117
 BL 76 114 124
 SC 89 108 144
 ST 94 104 128
 OF 116 115 135

APPENDIX H

SPSS REGRESSION PROGRAM LISTING

```

DATA LIST          FIXED/ 1 SQT 11-15 AFQT 17-18
                   CO 20-22 FA 24-26 MM 28-30 GM 32-34
                   CL 36-38 GT 40-42 EL 44-46 SC 48-50
                   ST 52-54 OF 56-58
VAR LABELS        SQT SKILL QUALIFICATION TEST/
                   AFQT AFQT PERCENTILE/
                   CO COMBAT/
                   FA FIELD ARTILLERY/
                   MM MOTOR MAINTENANCE/
                   GM GENERAL MAINTENANCE/
                   CL CLERICAL/
                   GT GENERAL TECHNICAL/
                   EL ELECTRONIC/
                   SC SURVEILLANCE, COMMUNICATIONS/
                   ST SKILLED TECHNICAL/
                   OF OPERATOR AND FOOD HANDLER
NEW REGRESSION    VARIABLES = SQT, CO TO OF/
DEP = SQT/STEPWISE/
RESIDUALS/CASEWISE = ALL/
SCATTER = (*RESID,*PRED)/SAVE = RESID PRED/

READ INPUT DATA
427310253 19 58 121 113 123 120 116 104 121 111 109 120
520683135 5 48 108 97 114 114 89 104 107 99 109 110
5277114668 8 70 105 103 112 113 95 117 113 100 107 108
539803606 3 66 106 102 109 107 109 109 105 111 112 114
553177646 6 26 98 91 91 94 89 89 87 94 94 93
560138625 17 44 114 112 107 99 106 94 101 108 92 101
560177925 5 28 101 98 102 105 78 95 103 84 99 93
573355886 11 70 114 105 112 109 98 117 110 104 113 116
4553393568 5 25 96 84 88 82 81 89 93 81 96 71
0393368355 14 95 118 125 115 118 122 124 125 128 126 105
092320162 6 21 90 65 91 89 84 87 85 81 64 82
096449837 3 60 106 111 107 99 99 112 112 101 111 88
1218445078 3 21 91 98 82 83 98 75 96 81 91 80
255043799 15 35 82 110 80 83 114 111 92 87 102 108
426150971 9 12 93 66 80 75 70 60 85 70 69 66
441707290 14 23 85 109 106 101 96 82 100 87 105 114
257044264 3 21 77 78 67 64 84 87 78 75 76 66
260252757 14 75 109 105 98 108 113 113 101 111 113 100
265538874 6 15 82 71 90 87 82 78 89 73 78 71
455594855 7 18 82 91 105 99 92 85 99 77 82 112
5088882557 5 77 95 125 97 113 123 126 116 116 128 100
050526224 8 25 85 92 85 87 84 80 90 84 93 88
220829366 2 41 83 79 73 78 82 98 81 91 87 71
231086870 6 53 82 106 109 109 101 115 106 100 105 110
423947244 7 44 91 84 81 65 96 96 81 91 75 80
5653374369 7 38 82 95 107 103 81 98 99 92 100 108
235665486 2 31 90 111 105 104 101 108 104 87 107 102
562254590 6 38 106 92 102 110 102 98 105 98 98 102
END INPUT DATA
FINISH

```

APPENDIX I

SPSS REGRESSION MODEL PROGRAM LISTING

```

DATA LIST          FIXED/ 1 SSAN 1-9 SQT 11-15 AFQT 17-18
                   CO 20-22 FA 24-26 MM 28-30 GM 32-34
                   CL 36-38 GT 40-42 EL 44-46 SC 48-50
                   ST 52-54 OF 56-58
VAR LABELS        SSAN  SOCIAL SECURITY NUMBER/
                   SQT  SKILL QUALIFICATION TEST/
                   AFQT  AFQT PERCENTILE/
                   CO    COMBAT/
                   FA    FIELD ARTILLERY/
                   MM    MOTOR MAINTENANCE/
                   GM    GENERAL MAINTENANCE/
                   CL    CLERICAL/
                   GT    GENERAL TECHNICAL/
                   EL    ELECTRONIC/
                   SC    SURVEILLANCE, COMMUNICATIONS/
                   ST    SKILLED TECHNICAL/
                   OF    OPERATOR AND FOOD HANDLER
COMPUTE           SQT=3+(0.085*GTT)
WRITE CASES      (F9.0,3X,F5.0) SSAN,SQT
READ INPUT DATA
15 58 121 113 123 120 116 104 121 111 109 120
13 48 108 113 114 114 89 104 107 99 109 110
21 70 105 103 112 113 95 117 113 100 107 108
20 66 106 102 109 107 109 109 105 111 112 114
6 26 58 91 91 89 89 87 94 94 93
12 44 114 112 107 99 106 94 101 108 92 101
5 28 101 98 102 105 78 95 103 84 99 93
23 70 114 109 112 109 98 117 110 104 113 116
5 25 96 84 88 81 89 93 31 96 71
28 95 118 125 115 118 122 124 125 128 126 105
6 21 90 65 91 89 84 87 85 81 64 82
15 60 106 111 107 99 99 112 112 101 111 88
3 21 91 98 82 83 38 75 96 81 91 80
5 35 82 110 80 83 114 111 92 87 102 108
1 12 93 66 80 75 70 60 85 70 69 66
7 23 85 109 106 101 96 82 100 87 105 114
7 21 77 78 67 64 84 87 78 75 76 66
18 75 109 105 98 108 113 113 101 111 113 100
6 15 82 71 90 87 82 78 89 73 78 71
7 18 82 91 105 99 92 85 99 77 82 112
19 77 95 125 97 113 123 126 116 116 128 100
10 25 85 92 85 87 84 80 90 84 93 88
12 41 83 79 73 78 82 98 81 91 87 71
16 53 82 106 109 109 101 115 106 100 105 110
13 44 91 84 81 65 96 81 91 75 80
10 38 82 95 107 103 81 98 99 92 100 108
9 31 90 111 105 104 101 108 104 87 107 102
11 38 106 92 102 110 102 98 105 98 98 102
END INPUT DATA
FINISH

```

APPENDIX J

SPSS REGRESSION PROGRAM OUTPUT

VARIABLE LIST NUMBER 1. LISTWISE DELETION OF MISSING DATA.

EQUATION NUMBER 1.

DEPENDENT VARIABLE.. SQT

'SKILL QUALIFICATION TEST

BEGINNING BLOCK NUMBER 1. METHOD: STEPWISE

VARIABLE(S) ENTERED ON STEP NUMBER 1.. CL CLERICAL

MULTIPLE R 0.47852
R SQUARE 0.22898
ADJUSTED R SQUARE 0.19932
STANDARD ERROR 4.21011

ANALYSIS OF VARIANCE
REGRESSION 1
RESIDUAL 26

SUM OF SQUARES
136.86306
460.85122

F = 7.72145 SIGNIF F = 0.0100

----- VARIABLES IN THE EQUATION -----

VARIABLE	B	SE B	BETA	T	SIG T
CL	0.16172	0.05820	0.47852	2.779	0.0100
(CONSTANT)	-7.73595	5.61678		-1.377	0.1802

VARIABLE
CO
FA
MM
GM
GTT
EL
SC
ST
OF

1SPSS BATCH SYSTEM

DEPENDENT VARIABLE.. SQT SKILL QUALIFICATION TEST

VARIABLE(S) ENTERED ON STEP NUMBER 2.. GT GENERAL TECHNICAL

MULTIPLE R 0.59073
R SQUARE 0.34897
ADJUSTED R SQUARE 0.29688
STANDARD ERROR 3.94529

ANALYSIS OF VARIANCE
REGRESSION 2
RESIDUAL 25

SUM OF SQUARES
208.58148
389.13281

SIGN IF $F = 0.0047$

SECRET

APPENDIX K

SPSS REGRESSION MODEL PROGRAM OUTPUT

SSAN	SQT
427310CE0	12
5206830C8	12
527714560	13
539803392	12
553177600	11
560138496	11
560177920	11
5735598C8	13
455393536	11
03933683E2	14
092520160	10
096449824	13
121545C72	9
255043792	12
4261504C0	8
4417C7264	10
257044256	10
260252784	13
265538E64	10
455548544	10
508882688	14
050526224	10
220829360	11
251086864	13
4239470C8	11
565374208	11
235565472	12
562254336	11

APPENDIX L

ASVAB PROGRAM LISTING

THIS PROGRAM READS IN THE SOLDIER DEMOGRAPHIC DATA, AND THE
ARMED SERVICES VOCATIONAL APPITUDE BATTERY (ASVAB) SUBTEST
RAW SCORES. THE RAW SCORES ARE CONVERTED INTO SUBTEST STANDARD
SCORES, AND THE ARMY CLASSIFICATION BATTERY (ACB) IS COMPUTED,
AND CONVERTED TO STANDARD SCORES. THE OUTPUT CONSISTS OF THE
SOLDIER DEMOGRAPHIC DATA, THE ASVAB RAW SCORES, THE ASVAB SUB-
TEST STANDARD SCORES, AND THE ACB COMPOSITE SCORES.

VARIABLES AND CONSTANTS

SSAN	SOCIAL SECURITY ACCOUNT NUMBER
DOBYY	YEAR OF BIRTH
DOBMM	MONTH OF BIRTH
DOBDD	DAY OF BIRTH
SEX	SEX
RACE	RACE
DCEYY	YEAR OF ENTRY
DOEMM	MONTH OF ENTRY
DOEDD	DAY OF ENTRY
ED	HIGHEST YEAR OF EDUCATION.
GRADE	ENTRY PAY GRADE.
MOS	MILITARY OCCUPATION SPECIALITY.
TEST	TEST FORM.
AFQT	ARMED FORCES QUALIFICATION PERCENTILE.
GI	GENERAL INFORMATION.
NG	GENERAL OPERATIONS.
AD	ATTENTION TO DETAIL.
WK	WORD KNOWLEDGE.
AR	ARITHMETIC REASONING.
SP	SPACE PERCEPTION.
MK	MATH KNOWLEDGE.
EI	ELECTRONIC INFORMATION.
MC	MECHANICAL COMPREHENSION.
GS	GENERAL SCIENCE.
SI	SHOP INFORMATION.
AI	AUTOMOTIVE INFORMATION.


```

C      INTEGER EL,SC,SI,OF,AFQT,MOS1,MOS2,J,K,A5(320,10),A67(111,10),
C      *A8910(281,10),SSAN1,SSAN2,SSAN3,END,ERROR
C      REAL CONVT,GIR,ADR,SPR,SIR,AIR,CMR,CAR,CER,CCR,GSR,ARR,WKR,PCR,
C      *NOR,CSR,ASR,WKR,MCR,EIR,VER
C      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      ALGORITHM ASVAB IS THE MASTER ALGORITHM. IT CONTAINS 7 SUB
C      ALGORITHMS. ALGORITHM ASVAB INITIALIZES A COUNTER (I), INPUTS
C      THE CONVERSION TABLES FOR ACB COMPOSITE SCORES, AND WRITES THE
C      HEADINGS FOR ALL FILES.
C      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      ALGORITHM ASVAB
C      CALL TABLE(A5,A67,A8910,I,J,K)
C      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      ALGORITHM INPUT SOLDIER DATA READS THE SOLDIER DEMOGRAPHIC DATA
C      AND THE ASVAB RAW SCORES.
C      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      ALGORITHM INPUT SOLDIER DATA
C      END = 0
C      CCNT INUE
C      CALL IN(SSAN1,SSAN2,SSAN3,DOBYY,DOBMM,DOBDD,SEX,RACE,DOEYY,
C      *DOEMM,DOEDDD,ED,GRADE,MOS1,MOS2,TEST,AFQT,RAW1,RAW2,RAW3,RAW4,RAW5,
C      *RAW6,RAW7,RAW8,RAW9,RAW10,RAW11,RAW12,RAW13,RAW14,RAW15,RAW16,END,
C      *DES,ETSY,ETSM,ETSD,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,
C      *CC,WK,PC,NG,CS,AS)

```



```

*GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,CC,WK,PC,NO,CS,AS)
  IF(ERROR .EQ. 1) GO TO 5
END ALGORITHM OUTPUT SOLDIER DATA

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ALGORITHM SUBTEST STANDARD SCORES CONVERTS THE SUBTEST RAW
SCORES TO SUBTEST STANDARD SCORES.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ALGORITHM SUBTEST STANDARD SCORES

CALL SSS(GI,AD,SP,SI,AI,CM,CA,CE,CC,GS,AR,WK,PC,NO,CS,AS,MK,MC,
*EI,VE,TEST,GI,SSS,AD,SSS,SP,SSS,SI,SSS,AI,SSS,CM,SSS,CASSS,CESSS,
*CCSSS,GS,SSS,AR,SSS,MK,SSS,PC,SSS,NO,SSS,CSSSS,ASSSS,MKSSS,MCSSS,
*EISSS,VESSS)

END ALGORITHM SUBTEST STANDARD SCORES

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ALGORITHM APTITUDE AREA COMPOSITES COMPUTES THE ACB COMPOSITE
SCORES.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ALGORITHM APTITUDE AREA COMPOSITES

CALL COMPI(GSSSS,ARSSS,WKSSS,PCSSS,NUSSS,CSSSS,ASSSS,MKSSS,
*MCSSS,EISSS,VESSS,GISSS,ADSSS,SPSSS,SISSS,AISSS,CMSSS,
*CASSS,CESSS,CCSSS,GS,AR,WK,PC,NO,CS,AS,MK,MC,EI,VE,GI,AD,SP,
*SI,AI,CM,CE,CA,CC,CO,FA,EL,OF,GM,MM,CL,GT,ST,SC,TEST)

```

[illegible]

116

[illegible]

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
SUBROUTINE IN(SSAN1,SSAN2,SSAN3,DOBY,DOBMM,DOBDD,SEX,RACE,DOEY,
*DOEM,DOEDD,ED,GRADE,MOS1,MOS2,TEST,AFQT,RAW1,RAW2,RAW3,RAW4,RAW5,
*RAW6,RAW7,RAW8,RAW9,RAW10,RAW11,RAW12,RAW13,RAW14,RAW15,RAW16,END,
*DES,ETSY,ETSM,ETSDD,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,
*CC,WK,PC,NC,CS,AS)
C
INTEGER SSAN1,SSAN2,SSAN3,DOBY,DOBMM,DOBDD,SEX,RACE,DOEY,
*DOEM,DOEDD,ED,GRADE,MOS1,MOS2,TEST,AFQT,RAW1,RAW2,RAW3,RAW4,RAW5,
*RAW6,RAW7,RAW8,RAW9,RAW10,RAW11,RAW12,RAW13,RAW14,RAW15,RAW16,END,
*DES,ETSY,ETSM,ETSDD,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,
*CC,WK,PC,NC,CS,AS
C
-----
C      INPUT DEMOGRAPHIC DATA AND RAW SCORES
C      -----
C
READ(4,500,END=1001) SSAN1,SSAN2,SSAN3,DOBY,DOBMM,DOBDD,SEX,RACE,
*DOEY,DOEM,DOEDD,ED,GRADE,MOS1,MOS2,TEST,AFQT,RAW1,RAW2,RAW3,
*RAW4,RAW5,RAW6,RAW7,RAW8,RAW9,RAW10,RAW11,RAW12,RAW13,RAW14,RAW15,
*RAW16,DES,ETSY,ETSM,ETSDD
C
-----
C      ASSIGN RAW SCORES TO VARIABLES
C      -----
C
C      ASVAB 8/9/10
C      -----
C
IF(TEST.LE.37) GO TO 6
GS=RAW1
AR=RAW2
WK=RAW3
PC=RAW4
NO=RAW5
CS=RAW6
AS=RAW7
MK=RAW8
MC=RAW9
EI=RAW10
VE=RAW11
GO TO 7

```

```

CCNT INUE
GI=RAW1
NO=RAW2
WK=RAW3
AR=RAW4
SPK=RAW5
EI=RAW6
MC=RAW7
GS=RAW8
AI=RAW9
CM=RAW10
CA=RAW11
CC=RAW12
CCNT INUE

```

```

CCNT INUE
RETURN
END = 1
RETURN

```

FORMAT (3A3,3I2,2I1,5I2,A4,A1,18I2,2X,A3,3I2)
END

SUBROUTINE VAL VALIDATES THE DEMOGRAPHIC DATA AND THE RAW SCORES. IF AN ERROR IS FOUND IT TERMINATES PROCESSING OF THE RECORD.

```

SUBROUTINE VAL(SSAN1,SSAN2,SSAN3,DOBY,DOBMM,DOBDD,SEX,RACE,
*DOEY,DOEMM,DOEDD,ED,TEST,G,AR,MK,MC,EI,VE,G,I,AD,SP,SI,AI,
*CM,CA,CE,CC,WK,PC,NO,CS,AS,EI,SY,EI,SMH,EI,SDD,I,ERROR)

```

INTEGER SSAN1,SSAN2,SSAN3,DOBYY,DOBMM,DOBDD,SEX,RACE,DOEYY,


```
*QOBMM,DOEDC,ED,TEST,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,
*CC,MK,PC,NO,CS,AS,ETSY,ETSM,ETSD,ET,ERROR
```

```
CC-----
CC  VALIDATE DATE OF BIRTH
CC-----
```

```
IF(DOBYV :GT. 31) GO TO 8
WRITE(9,510) I,SSAN1,SSAN2,SSAN3
WRITE(9,515)
ERROR = 1
CONTINUE
```

```
8CC-----
CC  VALIDATE SEX
CC-----
```

```
IF((SEX :EQ. 1) :OR. (SEX :EQ. 2)) GO TO 9
WRITE(9,510) I,SSAN1,SSAN2,SSAN3
WRITE(9,520)
ERROR = 1
CONTINUE
```

```
9CC-----
CC  VALIDATE RACE
CC-----
```

```
IF((RACE :EQ. 1) :OR. (RACE :EQ. 2) :OR. (RACE :EQ. 3)) GO TO 10
WRITE(9,510) I,SSAN1,SSAN2,SSAN3
WRITE(9,525)
ERROR = 1
CONTINUE
```

```
10CC-----
CC  VALIDATE DATE OF ENTRY
CC-----
```

```
IF(DOBYV :GT. 31) GO TO 11
WRITE(9,510) I,SSAN1,SSAN2,SSAN3
WRITE(9,530)
ERROR = 1
CONTINUE
```

```
11CC-----
CC  VALIDATE HIGHEST YEAR OF EDUCATION
CC-----
```

```
IF((ED :GE. 1) :OR. (ED :LE. 13)) GO TO 12
```

```

12      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,535)
        ERROR = 1
        CONTINUE
        -----
        VALIDATE TEST VERSION
        -----
        IF((TEST .GE. 35) .AND.(TEST .LE. 40)) GO TO 13
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,540)
        ERROR = 1
        CONTINUE
        -----
        VALIDATE SUBTEST RAW SCORES
        -----
        -----
        ASVAB 5/6/7
        -----
        IF(TEST .GT. 37) GO TO 30
        IF(GI .LE. 15) GO TO 14
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,545)
        ERROR = 1
        CONTINUE
        IF(NO .LE. 50) GO TO 15
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,550)
        ERROR = 1
        CONTINUE
        IF(AD .LE. 30) GO TO 16
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,555)
        ERROR = 1
        CONTINUE
        IF(WK .LE. 30) GO TO 17
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,560)
        ERROR = 1
        CONTINUE
        IF(AR .LE. 20) GO TO 18
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,565)
        ERROR = 1

```

18	CONTINUE IF(SP LE: 20) GO TO 19 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,570) ERROR = 1
19	CONTINUE IF(MK LE: 20) GO TO 20 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,575) ERROR = 1
20	CONTINUE IF(EI LE: 30) GO TO 21 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,580) ERROR = 1
21	CONTINUE IF(MC LE: 20) GO TO 22 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,585) ERROR = 1
22	CONTINUE IF(GS LE: 20) GO TO 23 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,590) ERROR = 1
23	CONTINUE IF(SI LE: 20) GO TO 24 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,595) ERROR = 1
24	CONTINUE IF(AI LE: 20) GO TO 25 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,600) ERROR = 1
25	CONTINUE IF(CM LE: 20) GO TO 26 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,605) ERROR = 1
26	CONTINUE IF(CA LE: 20) GO TO 27 WRITE(9,510) I, SSAN1, SSAN2, SSAN3 WRITE(9,610) ERROR = 1
27	CONTINUE IF(CE LE: 20) GO TO 28 WRITE(9,510) I, SSAN1, SSAN2, SSAN3

```

28      WRITE(9,615)
        ERROR = 1
        CONTINUE
        IF(CC LE 27) GO TO 29
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,620)
        ERROR = 1
        CONTINUE
29      GO TO 42
C-----
C      ASVAB 8/5/10
C-----
30      CCNTINUE
        IF(GS LE 25) GO TO 31
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,625)
        ERROR = 1
        CONTINUE
31      IF(AR LE 30) GO TO 32
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,630)
        ERROR = 1
        CONTINUE
32      IF(WK LE 35) GO TO 33
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,635)
        ERROR = 1
        CONTINUE
33      IF(PC LE 15) GO TO 34
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,640)
        ERROR = 1
        CONTINUE
34      IF(NO LE 50) GO TO 35
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,645)
        ERROR = 1
        CONTINUE
35      IF(CS LE 84) GO TO 36
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,650)
        ERROR = 1
        CONTINUE
36      IF(AS LE 25) GO TO 37
        WRITE(9,510) I,SSAN1,SSAN2,SSAN3
        WRITE(9,655)
        ERROR = 1

```

```

37      CONTINUE
      IF(MK LE 25) GO TO 38
      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
      WRITE(9,660)
      ERROR = 1
38      CONTINUE
      IF(MC LE 25) GO TO 39
      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
      WRITE(9,665)
      ERROR = 1
39      CONTINUE
      IF(EI LE 20) GO TO 40
      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
      WRITE(9,670)
      ERROR = 1
40      CONTINUE
      IF(VE LE 50) GO TO 41
      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
      WRITE(9,675)
      ERROR = 1
41      CONTINUE
42      CCCONTINUE
      C----- VALIDATE DATE OF SEPARATION -----
      C-----
43      IF(ETSY - EQ 00) GO TO 43
      IF(ETSY - GT 31) GO TO 43
      WRITE(9,510) I,SSAN1,SSAN2,SSAN3
      WRITE(9,680)
      ERROR = 1
      CCCONTINUE
      RETURN
510     FORMAT(//, ' ERROR IN RECORD NUMBER: ', I3, ' WITH SSAN: ', I3A3)
515     FORMAT(' ERROR: DATE OF BIRTH INCORRECT, ', I3, ' COLUMNS 10-15, ')
520     FORMAT(' ERROR: SEX INCORRECT, COLUMN 16, ')
525     FORMAT(' ERROR: RACE INCORRECT, COLUMN 17, ')
530     FORMAT(' ERROR: DATE OF ENTRY INCORRECT, COLUMNS 18-23, ')
535     FORMAT(' ERROR: HIGHEST YEAR OF EDUCATION INCORRECT, COLUMNS 24-
      *25, ')
540     FORMAT(' ERROR: TEST FORM INCORRECT, COLUMNS 33-34, ')
545     FORMAT(' ERROR: GI INCORRECT, COLUMNS 37-38, ')
550     FORMAT(' ERROR: NO INCORRECT, COLUMNS 39-40, ')
555     FORMAT(' ERROR: AD INCORRECT, COLUMNS 41-42, ')
560     FORMAT(' ERROR: WK INCORRECT, COLUMNS 43-44, ')
565     FORMAT(' ERROR: AR INCORRECT, COLUMNS 45-46, ')

```

[illegible]

```

C      WRITE(9,601) SSAN1,SSAN2,SSAN3,DOBYY,DOBMM,DOBDD
C      -----
C      PRINT SEX
C      -----
C
C      IF(SEX,NE,1) GO TO 45
C      WRITE(9,685)
C      GO TO 46
C      CCNTINUE
C      WRITE(9,690)
C      CONTINUE
C
C      -----
C      PRINT RACE
C      -----
C
C      IF(RACE,NE,1) GO TO 47
C      WRITE(9,695)
C      GO TO 50
C      IF(RACE,NE,2) GO TO 48
C      WRITE(9,700)
C      GO TO 50
C      CONTINUE
C      WRITE(9,705)
C      CONTINUE
C
C      -----
C      PRINT DATE OF ENTRY
C      -----
C
C      WRITE(9,710) DOEYY,DOEMM,DOEDD
C
C      -----
C      PRINT HIGHEST YEAR OF EDUCATION
C      -----
C
C      IF(ED,LT,1) GO TO 120
C      IF(ED,GT,13) GO TO 120
C      GO TO (55,60,65,70,75,80,85,90,95,100,105,110,115), ED
C      CONTINUE
C      WRITE(9,711) ED
C      GO TO 120
C      CONTINUE
C      WRITE(9,715) ED
C      GO TO 120
C      CONTINUE

```

70	WRITE(9,720) ED GO TO 120 CCNTINUE	
75	WRITE(9,725) ED GO TO 120 CCNTINUE	
80	WRITE(9,730) ED GO TO 120 CCNTINUE	
85	WRITE(9,735) ED GO TO 120 CCNTINUE	
90	WRITE(9,740) ED GO TO 120 CCNTINUE	
95	WRITE(9,745) ED GO TO 120 CCNTINUE	
100	WRITE(9,750) ED GO TO 120 CCNTINUE	
105	WRITE(9,755) ED GO TO 120 CCNTINUE	
110	WRITE(9,760) ED GO TO 120 CCNTINUE	
115	WRITE(9,765) ED GO TO 120 CCNTINUE	
120	WRITE(9,770) ED CCNTINUE	
	----- C PRINT GRADE C----- C	
125	IF(GRADE .LT. 1) GO TO 170 IF(GRADE .GT. 9) GO TO 170 GO TO (125,130,135,140,145,150,155,160,165),GRADE CCNTINUE	
130	WRITE(9,775) GRADE GO TO 170 CCNTINUE	
135	WRITE(9,780) GRADE GO TO 170 CCNTINUE WRITE(9,785) GRADE	


```

CC-----
CC      PRINT AFQT PERCENTILE
CC-----
CC
CC      WRITE(9,860) AFQT
CC-----
CC      PRINT SEPARATION PROGRAM DESIGNATOR
CC-----
CC
CC      WRITE(9,865) DES
CC-----
CC      PRINT DATE OF SEPARATION
CC-----
CC
CC      WRITE(9,870) ETSYY,ETSM,ETSDD
CC      WRITE(9,871)
CC-----
CC      PRINT ASVAB 5 RAW SCORES
CC-----
CC
CC      IF(TEST,NE,35) GO TO 205
CC      WRITE(9,874)
CC      WRITE(9,875) GI,NO,AD,WK,AR,SP,MK,EI,MC,GS,SI,AI
CC      GO TO 215
CC      CONTINUE
CC-----
CC      PRINT ASVAB 6/7 RAW SCORES
CC-----
CC
CC      IF(TEST,GT,37) GO TO 210
CC      WRITE(9,879)
CC      WRITE(9,880) GI,NO,AD,WK,AR,SP,MK,EI,MC,GS,SI,AI,CM,CA,CE,CC
CC      GO TO 215
CC      CONTINUE
CC-----
CC      PRINT ASVAB 8/9/10 RAW SCORES
CC-----
CC
CC      WRITE(9,884)
CC      WRITE(9,885) GS,AR,WK,PC,NO,CS,AS,MK,MC,EI,VE
CC      CONTINUE
CC      RETURN
CC-----

```

[illegible]


```

C-----
C  CCNVRT ASVAB 6/7
C-----
C
C  IF((TEST .NE. 36) .AND. (TEST .NE. 37)) GO TO 230
C    CMR=FLOAT(CM)
C    CHSSS=CCNVT(CMR,10.97,4.69)
C    CAR=FLOAT(CA)
C    CASSS=CCNVT(CAR,9.68,2.91)
C    CER=FLOAT(CE)
C    CESSS=CCNVT(CER,8.92,4.21)
C    CCR=FLOAT(CC)
C    CCESSS=CCNVT(CCR,15.51,4.89)
C  230 CONTINUE
C    RETURN
C  END
C
C  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    FUNCTION CCNVT CONVERTS THE SUBTEST RAW SCORES TO SUBTEST
C    STANDARD SCORES.
C
C  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    REAL FUNCTION CCNVT(RAW,XBAR,SIGMA)
C
C    -----
C    VARIABLES AND CONSTANTS
C    -----
C
C    RAW          SUBTEST RAW SCORE.
C    XBAR         SUBTEST POPULATION MEAN.
C    SIGMA        SUBTEST POPULATION STANDARD DEVIATION.
C    T           SUBTEST POPULATION TEMPORARY VARIABLE.
C    TT          TEMPORARY VARIABLE.
C    TTT         TEMPORARY VARIABLE.
C
C    -----
C    DECLARE VARIABLES
C    -----
C
C    INTEGER TT
C    REAL T,DIFF,TTT,XBAR,SIGMA,RAW
C
C

```

```

I=(10*(RA-XBAR)/SIGMA)+50.0
IT=IX(IT)
TT=FLOAT(TT)
DIFF=TT-TT
IF(DIFF-GE. 0.5) GO TO 10
CONVT=TT
GO TO 20
CCM INUE
CONVT=TT+1
CONT INUE
IF(CONVT:LI: 20) CONVT=20
IF(CONVT:GT: 80) CCNVT=80
RETURN
END

```

10
20

SUBROUTINE COMP COMPUTES THE ACB COMPOSITE SCCRES.

```

SUBROUTINE COMP(GSSSS,ARSSS,MKSSS,PCSSS,NOSSS,CSSSS,ASSSS,MKSSS,
*MCSSS,EISSS,VESSS,GISSS,ADSSS,SPSSS,SISSS,AISSS,CMSSS,
*CCSSS,CCSSS,GS,AR,MK,PC,NO,CS,AS,MK,MC,EI,VE,GT,AD,SP,
*SI,AI,CM,CE,CA,CC,CO,FA,EL,OF,GM,MM,CL,GT,ST,SC,TEST)
INTEGER GSSSS,ARSSS,MKSSS,PCSSS,NOSSS,CSSSS,ASSSS,MKSSS,
*MCSSS,EISSS,VESSS,GISSS,ADSSS,SPSSS,SISSS,AISSS,CMSSS,TEST,
*CCSSS,CCSSS,GS,AR,MK,PC,NO,CS,AS,MK,MC,EI,VE,GT,AD,SP,
*SI,AI,CM,CE,CA,CC,CO,FA,EL,OF,GM,MM,CL,GT,ST,SC

```

CCMPUTE ASVAB 5

```

IF( TEST:NE: 35) GO TO 235
CO=ARSSS+SISSS+SPSSS+ADSSS
FA=ARSSS+GISSS+MKSSS+EISSS
HM=MKSSS+SISSS+MCSSS+AISSS
CL=ARSSS+GS+MKSSS+ADSSS
GT=ARSSS+MKSSS+MCSSS+SISSS
EL=ARSSS+MKSSS+MCSSS+SPSSS
SC=ARSSS+MKSSS+MCSSS+SPSSS

```

```

ST=ARSSS+MKSSS+GSSSS
OF=GISSS+AIISSS
GO TO 243

```

```

C-----
C      COMPUTE ASVAB 6/7
C-----
C

```

```

235 IF((TEST .NE. 36) .AND. (TEST .NE. 37)) GO TO 240
      CO=AR+SI+SP+AD+CC
      FA=AR+SI+MK+EI+CA
      HM=MK+SI+EI+AI+CM
      GM=AR+GS+MC+AI
      CL=AR+MK+AD+CA
      GT=AR+MK
      EL=AR+EI+MC+SI+CE
      SC=AR+MK+MC+SP
      ST=AR+MK+GS
      OF=GI+AI+CA
      GO TO 245

```

```

C-----
C      COMPUTE ASVAB 8/9/10
C-----
C

```

```

240 CCNTINUE
      CO=ARSSS+ASSSSS+MKSSS+MCSSS+GSSSS
      FA=ARSSS+MKSSS+EISSS+VESSS+AISSS
      EL=ARSSS+EISSS+VESSS+MCSSS+GSSSS
      OF=NOSSS+VESSS+CSSSS+VESSS+GSSSS
      MM=NOSSS+CSSSS+EISSS+VESSS+GSSSS
      GM=MKSSS+EISSS+CSSSS+VESSS+GSSSS
      CL=NOSSS+CSSSS+MKSSS+ARSSS
      ST=VESSS+MKSSS
      GT=VESSS+ARSSS

```

```

245 CONTINUE
      RETURN
      END

```

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      SUBROUTINE ACB CONVERTS THE ACB SCORES INTO STANDARD SCORES.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```



```

CC      SUBROUTINE ACB(A5,A67,A8910,TEST,GT,GM,EL,CL,MM,SC,CO,FA,OF,ST)
C
CC      INTEGER A5(320,10),A67(111,10),A8910(281,10),GT,GM,EL,CL,MM,SC,
CC      *CO,FA,OF,ST,TEST
CC      -----
CC      CCNVERT ASVAB 5
CC      -----
CC      IF(TEST .NE. 35) GO TO 250
CC      GT=A5(GT,1)
CC      GM=A5(GM,2)
CC      EL=A5(EL,3)
CC      CL=A5(CL,4)
CC      MM=A5(MM,5)
CC      SC=A5(SC,6)
CC      CO=A5(CO,7)
CC      FA=A5(FA,8)
CC      OF=A5(OF,9)
CC      ST=A5(ST,10)
CC      GO TO 260
CC      -----
CC      CCNVERT ASVAB 6/7
CC      -----
CC250  IF((TEST .NE. 36) .AND.(TEST .NE. 37)) GO TO 255
CC      GT=A67(GT+1,1)
CC      GM=A67(GM+1,2)
CC      EL=A67(EL+1,3)
CC      CL=A67(CL+1,4)
CC      MM=A67(MM+1,5)
CC      SC=A67(SC+1,6)
CC      CO=A67(CO+1,7)
CC      FA=A67(FA+1,8)
CC      OF=A67(OF+1,9)
CC      ST=A67(ST+1,10)
CC      GO TO 260
CC      -----
CC      CCNVERT ASVAB 8/9/10
CC      -----
CC255  CCNT INJUE
CC      GT=A8910(GT,1)

```

```

GM=A8910(GM,2)
EL=A8910(EL,3)
CL=A8910(CL,4)
MM=A8910(MM,5)
SC=A8910(SC,6)
CO=A8910(CO,7)
FA=A8910(FA,8)
DF=A8910(DF,9)
ST=A8910(ST,10)

```

260

CGNT INUE

RETURN
END

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE OUT WRITES THE SUBTEST RAW SCORES TO FILE 11, THE
SUBTEST STANDARD SCORES TO FILES 9 AND 10, AND THE ACB COMPOSITE
SCORES TO FILES 9 AND 12.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

SUBROUTINE OUT(SSAN1,SSAN2,SSAN3,AFQT,GSSSS,ARSSS,MKSSS,MCSSS,
*EISSS,VESSS,GISSS,ADSSS,SPSSS,SISSS,AISSS,CMSSS,CASSS,CESSS,
*CCSSS,NOSSS,ASSS,MKSSS,PCSSS,TEST,CO,FA,MM,GM,CL,GT,
*EL,SC,ST,OF,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,CC,WK,
*PC,NO,CS,AS)

```

```

INTEGER SSAN1,SSAN2,SSAN3,AFQT,GSSSS,ARSSS,MKSSS,MCSSS,CSSSS,
*EISSS,VESSS,GISSS,ADSSS,SPSSS,SISSS,AISSS,CMSSS,CASSS,CESSS,
*CCSSS,NOSSS,ASSS,MKSSS,PCSSS,CO,FA,MM,GM,CL,GT,EL,SC,ST,OF,
*TEST,GS,AR,MK,MC,EI,VE,GI,AD,SP,SI,AI,CM,CA,CE,CC,WK,PC,NO,CS,AS

```

PRINT HEADINGS

WRITE(9,890)

PRINT SUBTEST RAW AND STANDARD SCORES

```

C-----
C      PRINT ASVAB 5
C-----

      IF(TEST,NE,35) GO TO 265
      WRITE(9,874) GISS,NOSSS,ADSSS,MKSSS,ARSSS,SPSSS,MKSSS,EISSS,
      * MCSSS,GSSSS,SISSS,AISSS
      * WRITE(10,876) SSAN1,SSAN2,SSAN3,GISSS,NOSSS,ADSSS,MKSSS,ARSSS,
      * SPSSS,MKSSS,EISSS,MCSSS,SISSS,AISSS
      * WRITE(11,876) SSAN1,SSAN2,SSAN3,GI,NO,AD,WK,AR,SP,MK,EI,MC,
      * GS,SI,AI
      GO TO 275

C-----
C      PRINT ASVAB 6/7
C-----

265  IF(TEST,GT,37) GO TO 270
      WRITE(9,879) GISS,NOSSS,ADSSS,MKSSS,ARSSS,SPSSS,MKSSS,EISSS,
      * MCSSS,GSSSS,SISSS,AISSS,CMSSS,CASSS,CSSSS
      * WRITE(10,881) SSAN1,SSAN2,SSAN3,GISSS,NOSSS,ADSSS,MKSSS,ARSSS,
      * SPSSS,MKSSS,EISSS,MCSSS,SISSS,AISSS,CMSSS,CASSS,CSSSS
      * CCSSS
      * WRITE(11,881) SSAN1,SSAN2,SSAN3,GI,NO,AD,WK,AR,SP,MK,EI,MC,
      * GS,SI,AI,CM,CA,CE,CC
      GO TO 275

C-----
C      PRINT ASVAB 8/9/10
C-----

270  CONTINUE
      WRITE(9,884) GSSSS,ARSSS,MKSSS,PCSSS,NOSSS,CSSSS,ASSSS,MKSSS,
      * MCSSS,EISSS,VESSS
      * WRITE(10,886) SSAN1,SSAN2,SSAN3,NOSSS,MKSSS,ARSSS,MKSSS,EISSS,
      * MCSSS,GSSSS,PCSSS,CSSSS,ASSSS,VESSS
      * WRITE(11,886) SSAN1,SSAN2,SSAN3,GS,AR,WK,PC,NO,CS,AS,MK,
      * MC,EI,VE
      CONTINUE

275  CONTINUE
C-----
C      PRINT ACB COMPOSITE SCORES
C-----

      WRITE(9,910)

```


BIBLIOGRAPHY

Graham, N., Introduction to Computer Science A Structured Approach, West Publishing Company, 1979.

Hussain, D. and Hussain, K.M., Information Processing Systems for Management, Richard D. Irwin, Incorporated, 1981.

Lipshutz, S. and Poe, A., Theory and Problems of Programming with FORTRAN, McGraw-Hill, 1978.

Nie, N., and others, Statistical Package for the Social Sciences, 2d Ed., McGraw-Hill, 1975.

Sperry Univac, Query Language Processor (QLP 1100) Level 4R1 Application Development Facilities, 1980.

Sperry Univac, Query Language Processor (QLP 1100) Level 4R1 Conversational Commands, 1980.

Sperry Univac, Query Language Processor (QLP 1100) Level 4R1 File and Application Definition Processors, 1980.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3. Commandant U.S. Army Infantry School Attn: ATSH-ES Fort Benning, Georgia 31905	1
4. LTC. Stephen Paek, Code 55Ph Naval Postgraduate School Monterey, California 93943	1
5. Captain Robert Forman 6550 Big Creek Parkway Parma Heights, Ohio 44130	1